```java
import java.time.YearMonth;

public class Person {

    protected String Name;
    protected String Address;
    protected String IC;
    protected String Phone;
    protected int Age;
    public Person(){};

    public Person(String name, String address, String ic, String phone){
        this.Name = name;
        this.Address = address;
        this.IC = ic;
        this.Phone = phone;
        this.Age = 0;
    }

    public void setPersonAge(int age){this.Age = age;}

    public String getPersonName(){return this.Name;}
    public String getPersonIC(){return this.IC;}
    public int getPersonAge(){return this.Age;}

    public int calcAge(){                //declare age
            int age;
            //get first 2 digit of ic
            String yearBorn = this.getPersonIC().substring(0,2);
            int yearIC = Integer.parseInt(yearBorn);
            //get the year born
            if(yearIC >= 0 && yearIC <= 23 ){
                yearIC = 2000 + yearIC;
            }
            else {
                yearIC = 1900 + yearIC;
            }
            //calculate the age by subtracting from current year
            age = YearMonth.now().getYear()-yearIC;
            //returning the age value
            setPersonAge(age);
            return age;

    }
}

```

```
1   public class Salesperson extends Person {
2
3       private String SPID;
4       private double SPcommissionRate;
5       private double CommissionEarned;
6       private Bill[] SPBill;
7       private int numBill;
8       private Customer[] servedCustomer;
9       private int numCustomer;
10
11      // constructor
12      public Salesperson(){};
13      public Salesperson(String name,String address,String ic, String phone, String SPID ,
        double commissionRate) {
14          super(name, address, ic, phone);
15          this.SPID = SPID;
16          this.SPcommissionRate = commissionRate;
17          this.CommissionEarned = 0;
18          this.numBill = 0;
19          this.numCustomer = 0;
20          this.SPBill = new Bill[100];
21          this.servedCustomer = new Customer[100];
22      }
23      // setter
24      public void setSalespersonData(String name,String address,String ic, String phone,
        String SPID , double commissionRate){
25          Name = name;
26          Address = address;
27          IC = ic;
28          Phone = phone;
29          this.SPID = SPID;
30          this.SPcommissionRate = commissionRate;
31          this.CommissionEarned = 0;
32          this.numBill = 0;
33          this.numCustomer = 0;
34          this.SPBill = new Bill[100];
35          this.servedCustomer = new Customer[100];
36      }
37
38      public String getSPID(){return this.SPID;}
39      public double getSPcommissionRate(){return this.SPcommissionRate;}
40
41
42      // processor
43         public void addBil(Bill bill){
44          if( numBill < SPBill.length){
45              SPBill[numBill] = bill;
46              numBill++;
47          }
48      }
49          public void addCustomer(Customer customer){
50              if(numCustomer < servedCustomer.length){
51                  servedCustomer[numCustomer] = customer;
52                  numCustomer++;
53              }
54          }
55
56          public void calcCommission(double earn){
57              CommissionEarned += earn;
58          }
59      // printer
60      public void PrintSalesPersonCommission(){
61          System.out.println("+---------------------------------+");
62          System.out.println("| Staff ID        : " + SPID);
63          System.out.println("| Name            : " + Name);
64          System.out.println("|");
65          System.out.println("| Commission rate : " + (getSPcommissionRate()*100) + "%") ;
66          System.out.println("| Commission      : RM" + CommissionEarned) ;
67          System.out.println("| Customer Served : " + numCustomer);
```

```
68          System.out.println("+--------------------------------+");
69      }
70
71      public void PrintSalesPersonInfo(){
72                  System.out.println("+--------------------------------+");
73                  System.out.println("| Name            : " + Name);
74                  System.out.println("| Address         : " + Address);
75                  System.out.println("| Phone           : " + Phone);
76                System.out.println("|");
77                  System.out.println("| Staff IC        : " + IC);
78                  System.out.println("| Staff ID        : " + SPID);
79                  System.out.println("| Commission rate : " + (getSPcommissionRate()*100)
                    + "%") ;
80                  System.out.println("| Commission      : RM" + CommissionEarned) ;
81                  System.out.println("| Customer Served : " + numCustomer);
82                  System.out.println("+--------------------------------+");
83      }
84  }
85
```

```
1
2   public class Customer extends Person {
3
4       private String password;
5       private int CustnumPurchasedCars;
6       private int numBill;
7       private Car[] CustpurchasedCars;
8       private Bill[] custBill;
9
10      // constructor
11      public Customer(){};
12
13      public Customer(String CustName, String CustAddress, String CustIC, String
        CustPhone, String Password) {
14          super(CustName, CustAddress, CustIC, CustPhone);
15          this.password = Password;
16          this.numBill = 0;
17          this.CustnumPurchasedCars = 0;
18          this.password = null;
19      }
20
21      // mutator
22         public void setCustPassword(String Passowrd){
23          this.password = Passowrd;
24      }
25      public void setCustomerData(String CustName, String CustAddress, String CustIC,
        String CustPhone,String Password) {
26          this.Name = CustName;
27          this.Address = CustAddress;
28          this.IC = CustIC;
29          this.Phone = CustPhone;
30          this.password = Password;
31          this.CustpurchasedCars = new Car[100];
32          this.custBill = new Bill[100];
33          this.numBill = 0;
34          this.CustnumPurchasedCars = 0;
35          this.Age = calcAge();
36      }
37      public String getCustPassword(){return this.password;}
38      public Bill[] getCustBill(){return custBill;}
39      public int getCustNumBill(){return numBill;}
40      public Car[] getCustPurchasedCars() {return CustpurchasedCars;}
41
42      // prcoessor
43      public void addPurchasedCar(Car car){
44          if( CustnumPurchasedCars < CustpurchasedCars.length){
45              CustpurchasedCars[CustnumPurchasedCars] = car;
46              CustnumPurchasedCars++;
47          }
48      }
49
50      public void addBil(Bill bill){
51          if( numBill < custBill.length){
52              custBill[numBill] = bill;
53              numBill++;
54          }
55      }
56
57       public void removeBill(Bill bill,int index){
58           for (int i = 0; i < custBill.length; i++) {
59              if (custBill[i] == custBill[index]) {
60                  index = i;
61                  break;
62              }
63          }
64              for(int i = index;i < custBill.length - 1 ;i++){
65                  custBill[i] = custBill[i + 1];
66          }
67          Bill[] newCustBill = new Bill[custBill.length - 1];
```

```
68          System.arraycopy(custBill, 0, newCustBill, 0, newCustBill.length);
69
70          custBill = newCustBill;
71
72          numBill--;
73      }
74      // printer
75      public String toString(){
76          String info = "+---------------------------------+"+
77                        "\n| Customer Name           : " + Name.toUpperCase() +
78                        "\n| Customer IC             : " + IC +
79                        "\n| Customer age            : " + calcAge() +
80                        "\n| Customer Address        : " + Address.toUpperCase() +
81                        "\n| Customer Phone number   : " + Phone +
82                        "\n| Number of Car Purchased : " + CustnumPurchasedCars + " cars";
83          return info;
84      }
85  }
86
```

```java
import java.text.DecimalFormat;

public class Bill {

    private String DateGenerated;

    private Salesperson SalesStaff;
    private Car SelectedCar;
    private Customer Customer;
    private int lengthOfLoan;
    private double totalPayment;
    private double monthlyPayment;
    private double paymentLeft;
    private double carDownpayment;

    DecimalFormat decimalFormat = new DecimalFormat("#.##");

    // default constructor
    public Bill(){}

    // normal constructor;
    public Bill(String Date , Salesperson salesStaff, Car selectCar, Customer
    customer,int YearPayment){
        this.DateGenerated = Date;
        this. SalesStaff = new Salesperson();
        this.SelectedCar = new Car();
        this.Customer = new Customer();
        this.lengthOfLoan = YearPayment;

    }

    // setter
    public void setTotalPayment(double totalPayment){
        this.totalPayment = totalPayment;
    }
    public void setMonthlyPayment(double MP){
        this.monthlyPayment = MP;
    }

    public void setBill(String Date , Salesperson salesperson, Car selectCar, Customer
    customer,int YearPayment){
        this.DateGenerated = Date;
        this. SalesStaff = salesperson;
        this.SelectedCar = selectCar;
        this.Customer = customer;
        this.lengthOfLoan = YearPayment;
        this.paymentLeft = payBill(0);
        this.totalPayment = calcTotalPayment(selectCar);
        this.carDownpayment = calcDownpayment(selectCar);
        this.monthlyPayment = calcMonthlyPayment();

    }

    // processor
    public int calcMonthOfPayment(){
        int month = 0;

        month = lengthOfLoan * 12;

        return month;
    }

    public double calcTotalPayment(Car car){
        double TotalPayment = 0.00;

        TotalPayment = car.calcCarPrice();

        setTotalPayment(TotalPayment);

```

```
68          return TotalPayment;
69      }
70      public double calcMonthlyPayment(){
71          double MPayment = 0.00;
72
73          MPayment = ((paymentLeft - carDownpayment ) / calcMonthOfPayment());
74
75          setMonthlyPayment(MPayment);
76
77          return MPayment;
78      }
79      public double payBill(double totalToPay){
80          paymentLeft = paymentLeft - totalToPay;
81
82          if(paymentLeft < 0){
83                      System.out.println("+---------------------------------+");
84                      System.out.println("| Balance of RM"+Math.abs(paymentLeft) + " will
                        be returned.");
85                      System.out.println("+---------------------------------+");
86
87          }
88
89          return totalPayment;
90
91      }
92      public double calcDownpayment(Car car){
93          double downpayment = 0.00;
94
95          downpayment = .1 * car.calcCarPrice();
96
97          paymentLeft = paymentLeft-carDownpayment;
98
99          return downpayment;
100     }
101     // getter
102     public Salesperson getsalesPerson(){return this.SalesStaff;}
103     public Car getSelectedCar(){return this.SelectedCar;}
104     public Customer getCustomer(){return this.Customer;}
105     public double getPaymentLeft(){return this.paymentLeft;}
106
107     // printer
108
109     public String SimplifiedBill(){
110         String str = "+---------------------------------+" +
111                     "\n|               BILL              |" +
112                     "\n+---------------------------------+" +
113                     "\n+  SALESPERSON NAME  : "+ getsalesPerson().getPersonName() +
114                     "\n+  CAR MODEL         : "+ getSelectedCar().getCarModel() +
115                     "\n+  CAR BRAND         : "+ getSelectedCar().getCarBrand() +
116                     "\n|                                 " +
117                     "\n|  PAYMENT LEFT DUE  : RM"+ paymentLeft +
118                     "\n+---------------------------------+";
119         return str;
120     }
121
122     public String toString(){
123       String str = "+---------------------------------+" +
124                     "\n|               BILL              |" +
125                     "\n+---------------------------------+" +
126                     "\n+  DATE GENERATED    : "+ DateGenerated +
127                     "\n|                                 " +
128                     "\n+  SALESPERSON NAME  : "+ getsalesPerson().getPersonName() +
129                     "\n+  CUSTOMER NAME     : "+ getCustomer().getPersonName() +
130                     "\n|                                 " +
131                     "\n+  CAR MODEL         : "+ getSelectedCar().getCarModel() +
132                     "\n+  CAR BRAND         : "+ getSelectedCar().getCarBrand() +
133                     "\n|                                 " +
134                     "\n|  TOTAL PAYMENT     : RM"+ totalPayment   +
135                     "\n|                                 " +
```

```
136                     "\n|  LOAN YEARS TIME   : "+lengthOfLoan+" years" +
137                     "\n|  LOAN MONTHLY TIME : "+calcMonthOfPayment()+" month"+
138                     "\n|                                      " +
139                     "\n|  CAR DOWNPAYMENT   : RM"+carDownpayment+
140                     "\n|  PAYMENT PER MONTH : RM"+monthlyPayment+
141                     "\n|                                      " +
142                     "\n|  PAYMENT LEFT DUE  : RM"+ paymentLeft +
143                     "\n+----------------------------------+";
144           return str;
145       }
146   }
```

```
public class Car {

    private String carBrand;
    private String carModel;
    private String carType;
    private double carPrice;

    // constructor
    public Car(){}
    public Car(String carBrand, String carModel, String carType, double carPrice) {
        this.carBrand = carBrand;
        this.carModel = carModel;
        this.carType = carType;
        this.carPrice = carPrice;
    }

    // getter
    public String getCarBrand(){return this.carBrand;}
    public String getCarModel(){return this.carModel;}
    public double getCarPrice(){return this.carPrice;}

    // processor
    public double calcCarPrice(){

        double price = 0.00;

        price = getCarPrice() + (getCarPrice() * .15);

        return price;
    }
    // printer
    @Override
    public String toString(){
        String info = "+---------------------------------+" +
                    "\n| Car Brand         : " + carBrand +
                    "\n| Car Model         : " + carModel +
                    "\n| Car Type          : " + carType +
                    "\n|" +
                    "\n| Car Price         : RM" + calcCarPrice() +
                    "\n+---------------------------------+";
        return info;
    }

    public String WriteString(){
        String write = carBrand + ";" + carModel + ";" + carType + ";" + carPrice;
        return write;
    }

}

```

```java
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

public class Dealership {

    private Salesperson[] SalesStaff;
    private Car[] CarInventory;
    private Customer[] CustomerData;
    private Bill[] bills;
    private int numSalesStaff;
    private int numCustomer;
    private int numCars;
    private int numBills;
    private Double TotalProfit;
    private Double TotalPaidCommission;
    private double ProfitRate;

    // constructor
    public Dealership(String DealershipName){
        this.SalesStaff = new Salesperson[4];  //4 is max number of staff
        this.CarInventory = new Car[100]; //100 car is limit
        this.CustomerData = new Customer[200]; //200 customer per system
        this.bills = new Bill[500]; //500 bill is limit per system run
        this.numSalesStaff = 0; //start with 0 staff
        this.numCustomer = 0;   //start with 0 customer
        this.numCars = 0; // start w    ith 0 car
        this.numBills = 0;
        this.TotalProfit = 0.00; // start with 0 profit
        this.TotalPaidCommission = 0.00; // start with 0 paid commission
        this.ProfitRate = 0.15; //15% profit rate
    }
    // getter
    public int getNumSalesStaff(){ return this.numSalesStaff;}
    public int getNumCustomer() { return this.numCustomer; }
    public int getNumCars() { return this.numCars; }
    public int getNumbills() { return this.numBills; }
    public double getProfitRate(){return this.ProfitRate;}
    public double getTotalProfit(){return this.TotalProfit;}
    public double getTotalPaidCommission(){return this.TotalPaidCommission;}
    public Salesperson[] getSalesStaff() {return this.SalesStaff;}
    public Car[] getCarInvetory(){return this.CarInventory;}
    public Customer[] getCustomerData(){return this.CustomerData;}

    // adding customer to database
    public void addCustomer(Customer customer){
                if(numCustomer < CustomerData.length){
                CustomerData[numCustomer] = customer;
            numCustomer++;
        }
        else{
            System.out.println("Cannot add more Customer. Limit is reached");
        }
    }
    public void addBill(Bill bill){
        if(numBills < bills.length){
            bills[numBills] = bill;
            numBills++;
        }
                else{
            System.out.println("Cannot add more bill. Limit is reached");
        }
    }
    // hiring staff
    public void hireSaleStaff(Salesperson salesperson){
        if(numSalesStaff < SalesStaff.length){
            SalesStaff[numSalesStaff] = salesperson;
            numSalesStaff++;
        }
        else{
```

```
70              System.out.println("Cannot add more staff. Limit is reached");
71          }
72      }
73       public void removeCarFromInventory(Car car,int index){
74           for (int i = 0; i < CarInventory.length; i++) {
75              if (CarInventory[i] == CarInventory[index]) {
76                  index = i;
77                  break;
78              }
79          }
80              for(int i = index;i < CarInventory.length - 1 ;i++){
81                  CarInventory[i] = CarInventory[i + 1];
82          }
83          Car[] newCarArray = new Car[CarInventory.length - 1];
84          System.arraycopy(CarInventory, 0, newCarArray, 0, newCarArray.length);
85          CarInventory = newCarArray;
86
87          numCars--;
88      }
89      public void addCarToInventory(Car car){
90                  if(numCars < CarInventory.length){
91                      CarInventory[numCars] = car;
92                      numCars++;
93              }
94                 else{
95                   System.out.println("Cannot add more car.Inventory is full.");
96                  }
97      }
98      public String GenerateDate(){
99          String date = "";
100         LocalDate currentDate = LocalDate.now();
101         DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
102         date = currentDate.format(formatter);
103
104         return date;
105     }
106     // calculating profit per car
107     public double calcProfit(Car car){
108         double profit = 0.00;
109         profit = car.calcCarPrice() - car.getCarPrice();
110         return profit;
111     }
112     // calculation commission per car that goes to salesPerson
113     public double calcCommission(Car car, Salesperson salesperson){
114         double Commission = 0.00;
115         Commission = (calcProfit(car) * salesperson.getSPcommissionRate()) ;
116         return Commission;
117     }
118     // calculating profit per car that goes to dealership
119     public double calcProfitToDealership(Car car, Salesperson salesperson){
120         double profit = 0.00;
121         profit = calcProfit(car) - calcCommission(car, salesperson);
122         return profit;
123     }
124     // calculating the total profit earn to Dealership;
125     public double calcTotalProfit(Car car, Salesperson salesperson){
126         TotalProfit += calcProfitToDealership(car, salesperson);
127         return TotalProfit;
128     }
129     // calculating totalCommission paid to SP;
130     public double calcTotalPaidCommission(Car car, Salesperson salesperson){
131         TotalPaidCommission += calcCommission(car, salesperson);
132         return TotalPaidCommission;
133     }
134     public void sellCar(int index,Car car, Salesperson salesperson, Customer
        customer,Bill bills){
135         calcCommission(car, salesperson);
136         addBill(bills);
137         calcProfit(car);
```

```
138             calcTotalProfit(car,salesperson);
139             calcTotalPaidCommission(car, salesperson);
140             removeCarFromInventory(car, index);
141             salesperson.calcCommission(calcCommission(car, salesperson));
142             System.out.println("+---------------------------------+");
143              System.out.println("| Car sold to " + customer.getPersonName() + " by " +
                 salesperson.getPersonName() +
144              " for RM" + car.calcCarPrice());
145              System.out.println("+ Revenue Gained : RM" + calcProfit(car));
146              System.out.println("+ Commission to " + salesperson.getPersonName()+"    : RM"
                 + calcCommission(car, salesperson));
147              System.out.println("| profit gain    : RM" +
                 (calcProfit(car)-calcCommission(car, salesperson)));
148              System.out.println("+---------------------------------+");
149              System.out.println(bills.toString());
150         }
151         public void printStaffCommission(Salesperson salesStaff[]){
152             for(int i = 0;i<salesStaff.length;i++){
153                 salesStaff[i].PrintSalesPersonCommission();
154             }
155             System.out.println("+---------------------------------+");
156              System.out.println("| Total Commission paid to Staff: RM" +
                 getTotalPaidCommission());
157             System.out.println("+---------------------------------+");
158         }
159         public void printStaffInformation(Salesperson salesStaff[]){
160             for(int i = 0;i<salesStaff.length;i++){
161                 salesStaff[i].PrintSalesPersonInfo();
162             }
163         }
164         public void PrintCarList(Car car[]){
165 
166             for(int i=0;i<numCars;i++){
167                     if(car[i] != null){
168                         System.out.println("| Car Number : " + (i+1));
169                         System.out.println(CarInventory[i]);
170                     }
171 
172         }
173             if(numCars == 0 ){
174             System.out.println("+---------------------------------+");
175                     System.out.println("|         UNFORTUNATELY           |");
176                     System.out.println("|     WE HAVE RAN OUT OF CAR      |");
177                     System.out.println("|     PLEASE COME AGAIN SOON      |");
178                     System.out.println("+---------------------------------+");
179                     }
180             }
181         public void printMainMenu(){
182             System.out.println("+---------------------------------+");
183             System.out.println("|  WELCOME TO KAMAL CAR SOLUTION  |");
184             System.out.println("+---------------------------------+");
185             System.out.println("|          [C] = Customer         |");
186             System.out.println("+                                 +");
187             System.out.println("|          [A] = Admin            |");
188             System.out.println("+                                 +");
189             System.out.println("|          [E] = exit             |");
190             System.out.println("+---------------------------------+");
191         }
192         public void printCustomerMenu(){
193                 System.out.println("+---------------------------------+");
194                  System.out.println("|               MENU              |");
195                  System.out.println("+---------------------------------+");
196                  System.out.println("|    [C] = Choose a car to buy    |");
197                  System.out.println("|    [I] = Car Invenotry          |");
198                  System.out.println("|    [B] = BIll sectcion          |");
199                  System.out.println("|    [U] = User Profile           |");
200                  System.out.println("|    [L] = Log Out                |");
201                  System.out.println("+---------------------------------+");
202         }
```

```
203       public void PrintStaffMainMenu(){
204                         // start of admin page
205           System.out.println("+---------------------------------+");
206           System.out.println("|           STAFF PAGE            |");
207           System.out.println("+---------------------------------+");
208           System.out.println("|     [P] = Total Profit          |");
209           System.out.println("|     [I] = Import Data           |");
210           System.out.println("+     [S] = Staff Management      +");
211           System.out.println("|     [C] = Car Management        |");
212           System.out.println("|     [A] = System Data           |");
213           System.out.println("|     [M] = Main Menu             |");
214           System.out.println("+---------------------------------+");
215       }
216       public void PrintStaffManagement(){
217                         // start of admin page
218           System.out.println("+---------------------------------+");
219           System.out.println("|     STAFF MANAGEMENT PAGE       |");
220           System.out.println("+---------------------------------+");
221           System.out.println("|     [C] = staff Commission      |");
222           System.out.println("|     [I] = Staff information     |");
223           System.out.println("|     [M] = Main Menu             |");
224           System.out.println("+---------------------------------+");
225       }
226       public void PrintCarManagement(){
227                         // start of admin page
228           System.out.println("+---------------------------------+");
229           System.out.println("|       CAR MANAGEMENT PAGE       |");
230           System.out.println("+---------------------------------+");
231           System.out.println("|     [A] = Add New Car           |");
232           System.out.println("|     [R] = Remove Car            |");
233           System.out.println("|     [I] = Car Inventory         |");
234           System.out.println("|     [M] = Main Menu             |");
235           System.out.println("+---------------------------------+");
236       }
237       public void PrintTotalProfit(){
238               System.out.println("+---------------------------------+");
239               System.out.println("|          TOTAL PROFIT           |");
240               System.out.println("+---------------------------------+");
241               System.out.println("| Profit Percentage : " + (getProfitRate()*100) + "%");
242               System.out.println("| Total Profit  : RM" + getTotalProfit());
243               System.out.println("+---------------------------------+");
244       }
245       public void printIOmenu(){
246           System.out.println("+---------------------------------+");
247           System.out.println("|              IMPORT             |");
248           System.out.println("+---------------------------------+");
249           System.out.println("|           [I] IMPORT            |");
250           System.out.println("|           [B] BACK TO MENU      |");
251           System.out.println("+---------------------------------+");
252       }
253       public void PrintSystemAnalytics(){
254           System.out.println("+---------------------------------+");
255           System.out.println("|        SYSTEM ANALYTICS         |");
256           System.out.println("+---------------------------------+");
257           System.out.println("|  NUMBER OF CUSTOMER           : " + numCustomer);
258           System.out.println("|  NUMBER OF STAFF              : " + numSalesStaff );
259           System.out.println("|  NUMBER OF CAR IN INVENTORY   : " + numCars);
260           System.out.println("|  EMPTY SPACE LEFT FOR CARS    : " + (100 - numCars));
261           System.out.println("|  NUMBER OF CAR SOLD           : " + numBills);
262           System.out.println("|  NMBER OF BILLS GENERATED     : " + numBills);
263           System.out.println("+---------------------------------+");
264       }
265   }
266
```

```java
import java.time.LocalDate;
import java.time.format.DateTimeFormatter;

public class Dealership {

    private Salesperson[] SalesStaff;
    private Car[] CarInventory;
    private Customer[] CustomerData;
    private Bill[] bills;
    private int numSalesStaff;
    private int numCustomer;
    private int numCars;
    private int numBills;
    private Double TotalProfit;
    private Double TotalPaidCommission;
    private double ProfitRate;

    // constructor
    public Dealership(String DealershipName){
        this.SalesStaff = new Salesperson[4];  //4 is max number of staff
        this.CarInventory = new Car[100]; //100 car is limit
        this.CustomerData = new Customer[200]; //200 customer per system
        this.bills = new Bill[500]; //500 bill is limit per system run
        this.numSalesStaff = 0; //start with 0 staff
        this.numCustomer = 0;   //start with 0 customer
        this.numCars = 0; // start w    ith 0 car
        this.numBills = 0;
        this.TotalProfit = 0.00; // start with 0 profit
        this.TotalPaidCommission = 0.00; // start with 0 paid commission
        this.ProfitRate = 0.15; //15% profit rate
    }
    // getter
    public int getNumSalesStaff(){ return this.numSalesStaff;}
    public int getNumCustomer() { return this.numCustomer; }
    public int getNumCars() { return this.numCars; }
    public int getNumbills() { return this.numBills; }
    public double getProfitRate(){return this.ProfitRate;}
    public double getTotalProfit(){return this.TotalProfit;}
    public double getTotalPaidCommission(){return this.TotalPaidCommission;}
    public Salesperson[] getSalesStaff() {return this.SalesStaff;}
    public Car[] getCarInvetory(){return this.CarInventory;}
    public Customer[] getCustomerData(){return this.CustomerData;}

    // adding customer to database
    public void addCustomer(Customer customer){
                if(numCustomer < CustomerData.length){
                CustomerData[numCustomer] = customer;
            numCustomer++;
        }
        else{
            System.out.println("Cannot add more Customer. Limit is reached");
        }
    }
    public void addBill(Bill bill){
        if(numBills < bills.length){
            bills[numBills] = bill;
            numBills++;
        }
                else{
            System.out.println("Cannot add more bill. Limit is reached");
        }
    }
    // hiring staff
    public void hireSaleStaff(Salesperson salesperson){
        if(numSalesStaff < SalesStaff.length){
            SalesStaff[numSalesStaff] = salesperson;
            numSalesStaff++;
        }
        else{
```

```
70              System.out.println("Cannot add more staff. Limit is reached");
71          }
72      }
73       public void removeCarFromInventory(Car car,int index){
74           for (int i = 0; i < CarInventory.length; i++) {
75              if (CarInventory[i] == CarInventory[index]) {
76                  index = i;
77                  break;
78              }
79          }
80              for(int i = index;i < CarInventory.length - 1 ;i++){
81                  CarInventory[i] = CarInventory[i + 1];
82          }
83          Car[] newCarArray = new Car[CarInventory.length - 1];
84          System.arraycopy(CarInventory, 0, newCarArray, 0, newCarArray.length);
85          CarInventory = newCarArray;
86
87          numCars--;
88      }
89      public void addCarToInventory(Car car){
90                  if(numCars < CarInventory.length){
91                      CarInventory[numCars] = car;
92                      numCars++;
93              }
94                 else{
95                   System.out.println("Cannot add more car.Inventory is full.");
96                  }
97      }
98      public String GenerateDate(){
99          String date = "";
100         LocalDate currentDate = LocalDate.now();
101         DateTimeFormatter formatter = DateTimeFormatter.ofPattern("yyyy-MM-dd");
102         date = currentDate.format(formatter);
103
104         return date;
105     }
106     // calculating profit per car
107     public double calcProfit(Car car){
108         double profit = 0.00;
109         profit = car.calcCarPrice() - car.getCarPrice();
110         return profit;
111     }
112     // calculation commission per car that goes to salesPerson
113     public double calcCommission(Car car, Salesperson salesperson){
114         double Commission = 0.00;
115         Commission = (calcProfit(car) * salesperson.getSPcommissionRate()) ;
116         return Commission;
117     }
118     // calculating profit per car that goes to dealership
119     public double calcProfitToDealership(Car car, Salesperson salesperson){
120         double profit = 0.00;
121         profit = calcProfit(car) - calcCommission(car, salesperson);
122         return profit;
123     }
124     // calculating the total profit earn to Dealership;
125     public double calcTotalProfit(Car car, Salesperson salesperson){
126         TotalProfit += calcProfitToDealership(car, salesperson);
127         return TotalProfit;
128     }
129     // calculating totalCommission paid to SP;
130     public double calcTotalPaidCommission(Car car, Salesperson salesperson){
131         TotalPaidCommission += calcCommission(car, salesperson);
132         return TotalPaidCommission;
133     }
134     public void sellCar(int index,Car car, Salesperson salesperson, Customer
        customer,Bill bills){
135         calcCommission(car, salesperson);
136         addBill(bills);
137         calcProfit(car);
```

```
138           calcTotalProfit(car,salesperson);
139           calcTotalPaidCommission(car, salesperson);
140           removeCarFromInventory(car, index);
141           salesperson.calcCommission(calcCommission(car, salesperson));
142           System.out.println("+--------------------------------+");
143            System.out.println("| Car sold to " + customer.getPersonName() + " by " +
              salesperson.getPersonName() +
144            " for RM" + car.calcCarPrice());
145            System.out.println("+ Revenue Gained : RM" + calcProfit(car));
146            System.out.println("+ Commission to " + salesperson.getPersonName()+"    : RM"
               + calcCommission(car, salesperson));
147            System.out.println("| profit gain    : RM" +
               (calcProfit(car)-calcCommission(car, salesperson)));
148            System.out.println("+--------------------------------+");
149            System.out.println(bills.toString());
150       }
151       public void printStaffCommission(Salesperson salesStaff[]){
152           for(int i = 0;i<salesStaff.length;i++){
153               salesStaff[i].PrintSalesPersonCommission();
154           }
155           System.out.println("+--------------------------------+");
156            System.out.println("| Total Commission paid to Staff: RM" +
               getTotalPaidCommission());
157           System.out.println("+--------------------------------+");
158       }
159       public void printStaffInformation(Salesperson salesStaff[]){
160           for(int i = 0;i<salesStaff.length;i++){
161               salesStaff[i].PrintSalesPersonInfo();
162           }
163       }
164       public void PrintCarList(Car car[]){
165
166           for(int i=0;i<numCars;i++){
167                   if(car[i] != null){
168                       System.out.println("| Car Number : " + (i+1));
169                       System.out.println(CarInventory[i]);
170                   }
171       }
172           if(numCars == 0 ){
173           System.out.println("+--------------------------------+");
174                   System.out.println("|         UNFORTUNATELY          |");
175                   System.out.println("|     WE HAVE RAN OUT OF CAR     |");
176                   System.out.println("|     PLEASE COME AGAIN SOON     |");
177                   System.out.println("+--------------------------------+");
178                   }
179           }
180       public void printMainMenu(){
181           System.out.println("+--------------------------------+");
182           System.out.println("|  WELCOME TO KAMAL CAR SOLUTION  |");
183           System.out.println("+--------------------------------+");
184           System.out.println("|          [C] = Customer        |");
185           System.out.println("+                                +");
186           System.out.println("|          [A] = Admin           |");
187           System.out.println("+                                +");
188           System.out.println("|          [E] = exit            |");
189           System.out.println("+--------------------------------+");
190       }
191       public void printCustomerMenu(){
192                   System.out.println("+--------------------------------+");
193                    System.out.println("|              MENU              |");
194                    System.out.println("+--------------------------------+");
195                    System.out.println("|    [C] = Choose a car to buy   |");
196                    System.out.println("|    [I] = Car Invenotry         |");
197                    System.out.println("|    [B] = BIll sectcion         |");
198                    System.out.println("|    [U] = User Profile          |");
199                    System.out.println("|    [L] = Log Out               |");
200                    System.out.println("+--------------------------------+");
201       }
202       public void PrintStaffMainMenu(){
```

```
                        // start of admin page
        System.out.println("+---------------------------------+");
        System.out.println("|           STAFF PAGE            |");
        System.out.println("+---------------------------------+");
        System.out.println("|     [P] = Total Profit          |");
        System.out.println("|     [I] = Import Data           |");
        System.out.println("+     [S] = Staff Management      +");
        System.out.println("|     [C] = Car Management        |");
        System.out.println("|     [A] = System Data           |");
        System.out.println("|     [M] = Main Menu             |");
        System.out.println("+---------------------------------+");
    }
    public void PrintStaffManagement(){
                        // start of admin page
        System.out.println("+---------------------------------+");
        System.out.println("|     STAFF MANAGEMENT PAGE       |");
        System.out.println("+---------------------------------+");
        System.out.println("|     [C] = staff Commission      |");
        System.out.println("|     [I] = Staff information     |");
        System.out.println("|     [M] = Main Menu             |");
        System.out.println("+---------------------------------+");
    }
    public void PrintCarManagement(){
                        // start of admin page
        System.out.println("+---------------------------------+");
        System.out.println("|       CAR MANAGEMENT PAGE       |");
        System.out.println("+---------------------------------+");
        System.out.println("|     [A] = Add New Car           |");
        System.out.println("|     [R] = Remove Car            |");
        System.out.println("|     [I] = Car Inventory         |");
        System.out.println("|     [M] = Main Menu             |");
        System.out.println("+---------------------------------+");
    }
    public void PrintTotalProfit(){
            System.out.println("+---------------------------------+");
            System.out.println("|          TOTAL PROFIT           |");
            System.out.println("+---------------------------------+");
            System.out.println("| Profit Percentage : " + (getProfitRate()*100) + "%");
            System.out.println("| Total Profit  : RM" + getTotalProfit());
            System.out.println("+---------------------------------+");
    }
    public void printIOmenu(){
        System.out.println("+---------------------------------+");
        System.out.println("|              IMPORT             |");
        System.out.println("+---------------------------------+");
        System.out.println("|           [I] IMPORT            |");
        System.out.println("|           [B] BACK TO MENU      |");
        System.out.println("+---------------------------------+");
    }
    public void PrintSystemAnalytics(){
        System.out.println("+---------------------------------+");
        System.out.println("|        SYSTEM ANALYTICS         |");
        System.out.println("+---------------------------------+");
        System.out.println("|  NUMBER OF CUSTOMER           : " + numCustomer);
        System.out.println("|  NUMBER OF STAFF              : " + numSalesStaff );
        System.out.println("|  NUMBER OF CAR IN INVENTORY   : " + numCars);
        System.out.println("|  EMPTY SPACE LEFT FOR CARS    : " + (100 - numCars));
        System.out.println("|  NUMBER OF CAR SOLD           : " + numBills);
        System.out.println("|  NMBER OF BILLS GENERATED     : " + numBills);
        System.out.println("+---------------------------------+");
    }
}
import java.util.*;
import java.io.*;

public class CarDealershipApp {

    private static final String CAR_FILE = "Car.txt";
    public static int Size = 100;
```

```
272       public static void main(String[] args) throws Exception{
273           // Create a new Dealership
274           Dealership dealership = new Dealership("KAMAL CAR SOLUTION SDN BHD");
275
276           Customer[] customer = new Customer[Size];
277           Salesperson[] SalesStaff = new Salesperson[3];
278           Car[] cars = new Car[Size];
279
280           // add salestaff
281           SalesStaff[0] = new Salesperson("SITI BINTI HALIM","NO 30, JALAN RIMAU 16, TAMAN
              RIMAU,43650, BANDAR BARU BANGI","650525035450","011-2345 6789","SP01", 0.1);
282           SalesStaff[1] = new Salesperson("AHMAD BIN MUHAMMAD","NO 25, JALAN EMPAYAN
              4,TAMAN EMPAYAN,43650, BANDAR BARU BANGI","650525035450","011-2345 6789","SP02",
              0.15);
283           SalesStaff[2] = new Salesperson("ABU BIN KASIM","NO 1, JALAN EMPAYAN 20, TAMAN
              EMPAYAN,43650, BANDAR BARU BANGI","650525035450","012-4363 5764","SP03", 0.12);
284
285           for(int i = 0; i < SalesStaff.length; i++){
286               dealership.hireSaleStaff(SalesStaff[i]);
287           }
288           // get customer data
289           Scanner input = new Scanner(System.in);
290           String optionTemp;
291           char option;
292
293           readFile(cars, dealership);
294           do{
295
296           dealership.printMainMenu();
297
298           System.out.print("| Choose one to continue : ");
299            optionTemp = input.next();
300           option = optionTemp.toUpperCase().charAt(0);
301
302           if(option == 'C'){
303               Scanner customerin = new Scanner(System.in);
304               String optionCust = "E";
305               while(optionCust.toUpperCase().charAt(0) != 'M'){
306                System.out.println("+---------------------------------+");
307                System.out.println("|              LOGIN              |");
308                System.out.println("+---------------------------------+");
309                System.out.println("|    [L] = Login to account       |");
310                System.out.println("|    [R] = Register an account    |");
311                System.out.println("|    [M] = Back to menu           |");
312                System.out.println("+---------------------------------+");
313               System.out.print("| Choose : ");
314                optionCust = customerin.next();
315                if(optionCust.toUpperCase().charAt(0) == 'R'){
316                   RegisterCustomerProcess(customer, dealership);
317                   }
318                else if(optionCust.toUpperCase().charAt(0) == 'L'){
319                   int attempt = 3;
320                   boolean success = false;
321                   Scanner customerLogin = new Scanner(System.in);
322                   if(dealership.getNumCustomer() == 0){
323                   System.out.println("+---------------------------------+");
324                   System.out.println("|    ERROR, NO CUSTOMER DATA      |");
325                   System.out.println("|    PLEASE CREATE AN ACCOUNT     |");
326                   System.out.println("+---------------------------------+");
327                   break;
328                   }
329                   else{
330                   System.out.println("+---------------------------------+");
331                   System.out.println("|              LOGIN              |");
332                   System.out.println("+---------------------------------+");
333                   }
334
335                   while(attempt != 0 && success == false){
336
```

```
337                     System.out.print("| IC NUMBER [without -] : ");
338                     String IC = customerLogin.nextLine();
339                     System.out.print("| PASSWORD :");
340                     String pass = customerLogin.nextLine();
341 
342                     int indexofCustomer = 0;
343                 for(int j = 0 ; j<dealership.getNumCustomer();j++){
344 
345                 if(customer[j].getCustPassword().equals(pass) &&
                    customer[j].getPersonIC().equals(IC)){
346                     String optionCust2 = "Meee";
347                         success = true;
348                         System.out.println("+---------------------------------+");
349                         System.out.println("|             SUCCESS             |");
350                         System.out.println("+---------------------------------+");
351                         indexofCustomer = j;
352                          do{
353                     Scanner customerMenu = new Scanner(System.in);
354                     dealership.printCustomerMenu();
355                    System.out.print("| Choose : ");
356                 optionCust2 = customerMenu.nextLine();
357                 if(optionCust2.toUpperCase().charAt(0) == 'C'){
358                 if(dealership.getNumCars() == 0){
359                 System.out.println("+---------------------------------+");
360                     System.out.println("|         UNFORTUNATELY           |");
361                     System.out.println("|     WE HAVE RAN OUT OF CAR      |");
362                     System.out.println("|     PLEASE COME AGAIN SOON      |");
363                     System.out.println("+---------------------------------+");
364         }
365         else{
366         System.out.println("+---------------------------------+");
367         System.out.println("|          Available Car          |");
368         System.out.println("+---------------------------------+");
369 
370         dealership.PrintCarList(cars);
371          while(true){
372         System.out.println("+---------------------------------+");
373         System.out.println("|           Choose a car          |");
374         System.out.println("+---------------------------------+");
375         System.out.print("|  car number : ");
376         int carChoice = customerin.nextInt();
377 
378         if(carChoice >=1 && carChoice <= cars.length){
379             Car selectedCar = cars[ carChoice - 1];
380 
381             //code index to randomly assign a salesperson to a customer
382             int startIndex = 1;
383             int endIndex = dealership.getNumSalesStaff();
384             Random random = new Random();
385             int randomIndex = random.nextInt(endIndex - startIndex ) + startIndex;
386 
387             System.out.println("+---------------------------------+");
388             System.out.println("| Mr." + SalesStaff[randomIndex].getPersonName() + "(
                ID: " + SalesStaff[randomIndex].getSPID() + " ) " + "will handle\n| your
                purchase transaction.");
389             System.out.println("+---------------------------------+");
390             System.out.println("| do you wish to proceed and buy the following vehicle
                :");
391             System.out.println(selectedCar);
392             System.out.print("| Answer Y/N : ");
393             String answer = customerin.next();
394             SalesStaff[randomIndex].addCustomer(customer[indexofCustomer]);
395             if(answer.toUpperCase().charAt(0) == 'Y'){
396                 System.out.println("+---------------------------------+");
397                 System.out.println("|       Choose Loan Length        |");
398                 System.out.println("+---------------------------------+");
399                 System.out.println("|   [6] = 6 Year Loan             |");
400                 System.out.println("|   [9] = 9 Year Loan             |");
401                 System.out.println("+---------------------------------+");
```

```
402                 System.out.print("| Answer : ");
403                 String paymentAnswer = customerin.next();
404                 int loanLength = 0;
405 
406                 Bill bills = new Bill();
407                 String date = dealership.GenerateDate();
408 
409                 if(paymentAnswer.charAt(0) == '6'){
410                 loanLength = 6;
411                 bills.calcTotalPayment(selectedCar);
412                 bills.setBill(date, SalesStaff[randomIndex], selectedCar,
                    customer[indexofCustomer],loanLength);
413                 bills.payBill(bills.calcDownpayment(selectedCar));
414                 customer[indexofCustomer].addPurchasedCar(cars[carChoice - 1]);
415                 SalesStaff[randomIndex].addBil(bills);
416                 customer[indexofCustomer].addBil(bills);
417                 deleteCarData(cars, dealership, (carChoice-1));
418                 dealership.sellCar(carChoice-1,cars[carChoice-1],
                    SalesStaff[randomIndex], customer[indexofCustomer],bills);
419 
420                 System.out.println(dealership.getNumbills());
421                 break;
422             }
423                 else if(paymentAnswer.charAt(0) == '9'){
424                 loanLength = 9;
425                 bills.calcTotalPayment(selectedCar);
426                 bills.setBill(date, SalesStaff[randomIndex], selectedCar,
                    customer[indexofCustomer],loanLength);
427                 bills.payBill(bills.calcDownpayment(selectedCar));
428                 customer[indexofCustomer].addPurchasedCar(cars[carChoice - 1]);
429                 SalesStaff[randomIndex].addBil(bills);
430                 customer[indexofCustomer].addBil(bills);
431                 deleteCarData(cars, dealership, (carChoice-1));
432                 dealership.sellCar(carChoice-1,cars[carChoice-1],
                    SalesStaff[randomIndex], customer[indexofCustomer],bills);
433                 System.out.println(dealership.getNumbills());
434                 break;
435                 }
436 
437             }
438             else{
439                 System.out.println("+---------------------------------+");
440                 System.out.println("|       Choose another car ?      |");
441                 System.out.println("+---------------------------------+");
442                 System.out.print("| Answer Y/N : ");
443                 answer = customerin.next();
444                 if(answer.toUpperCase().charAt(0) == 'Y'){
445                     dealership.PrintCarList(cars);
446                 }
447                 else{
448                     System.out.println("+---------------------------------+");
449                     System.out.println("|    Thank you for using us !     |");
450                     System.out.println("+---------------------------------+");
451                     break;
452                 }
453 
454             }
455 
456         }
457         else {
458             System.out.println("|  Invalid car choice.");
459         }
460     }
461 
462         }
463         // opt user to choose which car to buyE
464             }
465             else if(optionCust2.toUpperCase().charAt(0) == 'I'){
466             // display all available car list
```

```
467             System.out.println("+---------------------------------+");
468             System.out.println("|          Available Car          |");
469             System.out.println("+---------------------------------+");
470
471             dealership.PrintCarList(cars);
472             }
473             else if(optionCust2.toUpperCase().charAt(0) == 'B'){
474
475             Scanner optionbillCust = new Scanner(System.in);
476
477             int repeat5 = 0;
478             do{
479              System.out.println("+---------------------------------+");
480              System.out.println("|            BILL MENU            |");
481              System.out.println("+---------------------------------+");
482              System.out.println("|    [P] = Pay Bill               |");
483              System.out.println("|    [L] = List of Bill           |");
484              System.out.println("|    [B] = Back to Menu           |");
485              System.out.println("+---------------------------------+");
486             System.out.print("| Choose : ");
487             String optionBill = optionbillCust.next();
488             Bill[] customerbill = customer[indexofCustomer].getCustBill();
489             if(optionBill.toUpperCase().charAt(0) == 'P'){
490                     int chosenBill = 0;
491                     double amountToPay = 0;
492                 if(customer[indexofCustomer].getCustNumBill() == 0) {
493                         System.out.println("| NO BILL AVAILABLE");
494                 }
495                 else {
496                     for(int i =0;i<customer[indexofCustomer].getCustNumBill(); i++){
497
498
499                     Bill bill = customerbill[i];
500                     System.out.println("| BILL NUMBER " + (i+1));
501                     System.out.println(bill.SimplifiedBill());
502
503                 }
504                     System.out.print("| Choose number of bills : ");
505                     chosenBill = optionbillCust.nextInt();
506                     int indexofBill = chosenBill - 1;
507                     if(indexofBill > 99 || indexofBill < 0){
508                         System.out.println("| ERROR : Bill Not Found.");
509                         System.out.println("| Going back to menu.");
510                         break;
511                     }
512                     Bill bill = customerbill[indexofBill];
513
514                     if(bill != null){
515                     System.out.println("+---------------------------------+");
516                     System.out.println("|          BILL NUMBER "+chosenBill+"
                        |");
517                     System.out.println("|         HAS BEEN CHOSEN         |");
518                     System.out.println("+---------------------------------+");
519                     System.out.println(bill);
520                     System.out.println("| How much would you like to pay ?");
521                     System.out.print("| RM");
522                     amountToPay = optionbillCust.nextDouble();
523                     bill.payBill(amountToPay);
524                     System.out.println("+---------------------------------+");
525                     System.out.println("|        PAYMENT SUCCESSFUL       |");
526                     System.out.println("+---------------------------------+");
527
528
529                     if(bill.getPaymentLeft() == 0 || bill.getPaymentLeft() < 0){
530                         customer[indexofCustomer].removeBill(bill, indexofBill);
531                     System.out.println("+---------------------------------+");
532                     System.out.println("|         BILL FULLY PAID         |");
533                     System.out.println("|         TERMINATING BILL        |");
534                     System.out.println("+---------------------------------+");
```

```
                    }
                    }
                    else{
                        System.out.println("| ERROR, NO BILL FOUND");
                    }
                }

            }
           else if(optionBill.toUpperCase().charAt(0) == 'L'){
                for(int i = 0; i < customer[indexofCustomer].getCustNumBill(); i++){
                    Bill bill = customerbill[i];
                    System.out.println(bill.SimplifiedBill());
                }
                if(customer[indexofCustomer].getCustNumBill() == 0) {
                        System.out.println("| NO BILL AVAILABLE");
                }
            }
            else if(optionBill.toUpperCase().charAt(0) == 'B'){
                repeat5 = 1;
            }
            }while(repeat5 == 0);

            }
            else if(optionCust2.toUpperCase().charAt(0) == 'U'){
                System.out.println(customer[indexofCustomer]);
            }
            else if(optionCust2.toUpperCase().charAt(0) == 'L'){
                // memang kosong sebab taknak buat apa apa sebenarnya
            }
            else{
                System.out.println("| ERROR : Wrong Code Entered");
            }



            }while(optionCust2.toUpperCase().charAt(0) != 'L');

                }
            else{
                System.out.println("+---------------------------------+");
                System.out.println("|              ERROR              |");
                System.out.println("|     YOU HAVE "+(attempt-1)+" ATTEMPT LEFT     |");
                System.out.println("+---------------------------------+");
                    attempt--;
            }
            }

        }
         }
        }
    }
        else if(option == 'A'){
            Scanner admin = new Scanner(System.in);
            int attemptleft = 3;
            boolean success = false;
            System.out.println("+---------------------------------+");
            System.out.println("|           ADMIN PAGE            |");
            System.out.println("+---------------------------------+");
            System.out.println("|          PLEASE LOGIN           |");
            System.out.println("|        YOU HAVE 3 ATTEMPT       |");
            System.out.println("+---------------------------------+");
            while(attemptleft != 0 && success == false){
                System.out.print("| Username : ");
                String username = admin.nextLine();
                System.out.print("| Password : ");
                String Password = admin.nextLine();
            if(username.equals("staffKamalSolution") && Password.equals("admin123")){
                success = true;
            int repeat2 = 0;
```

```
604                 while(repeat2 != 1){
605                 dealership.PrintStaffMainMenu();
606                 System.out.print("| Choose one to continue : ");
607                 String option2 = input.next();
608 
609                 if(option2.toUpperCase().charAt(0) == 'P'){
610                     dealership.PrintTotalProfit();
611                 }
612                 else if(option2.toUpperCase().charAt(0) == 'I'){
613 
614                     Scanner adminIO = new Scanner(System.in);
615 
616                     String optionIO = "C";
617                     while(optionIO.toUpperCase().charAt(0) != 'B'){
618                     dealership.printIOmenu();
619                     System.out.print("| Choose : ");
620                     optionIO = adminIO.next();
621 
622                     if(optionIO.toUpperCase().charAt(0) == 'I'){
623                             importCar(cars, dealership,"ImportCar.txt");
624                     }else{
625 
626                     }
627                 }
628                 }
629                 else if(option2.toUpperCase().charAt(0) == 'C'){
630 
631                     Scanner Manage = new Scanner(System.in);
632                     String carOption = "G";
633 
634                     while(carOption.toUpperCase().charAt(0) != 'M'){
635                     dealership.PrintCarManagement();
636                     System.out.print("| Choose : ");
637                     carOption = Manage.nextLine();
638                      if(carOption.toUpperCase().charAt(0) == 'A'){
639                     addNewCar(cars, dealership);
640                     }
641 
642                     else if(carOption.toUpperCase().charAt(0) == 'R'){
643                         RemoveCar(cars, dealership);
644                     }
645                     else if(carOption.toUpperCase().charAt(0) == 'I'){
646                     System.out.println("+---------------------------------+");
647                     System.out.println("|            CAR STOCK            |");
648                     System.out.println("+---------------------------------+");
649                     dealership.PrintCarList(cars);
650                 }
651                 else {
652 
653                 }
654                 }
655 
656 
657                 }
658                 else if(option2.toUpperCase().charAt(0) == 'S'){
659                 Scanner Manage = new Scanner(System.in);
660                 String staffOption = "G";
661                 while(staffOption.toUpperCase().charAt(0) != 'M'){
662 
663                 dealership.PrintStaffManagement();
664                 System.out.print("| Choose : ");
665                 staffOption = Manage.nextLine();
666 
667                     if(staffOption.toUpperCase().charAt(0) == 'C'){
668                         dealership.printStaffCommission(SalesStaff);
669                     }
670                     else if(staffOption.toUpperCase().charAt(0) == 'I'){
671                         dealership.printStaffInformation(SalesStaff);
672                     }
```

```
673                 else{
674                 }
675             }
676 
677             }
678             else if(option2.toUpperCase().charAt(0) == 'A'){
679                 dealership.PrintSystemAnalytics();
680             }
681             else if(option2.toUpperCase().charAt(0) == 'M'){
682                 repeat2++;
683             }
684             }
685         }
686             else {
687                 attemptleft--;
688                 System.out.println("+---------------------------------+");
689                 System.out.println("|              ERROR              |");
690                 System.out.println("|     YOU HAVE "+attemptleft+" ATTEMPT LEFT     |");
691                 System.out.println("+---------------------------------+");
692             }
693 
694         }
695         }
696         else if(option == 'E'){
697             System.out.println("+---------------------------------+");
698             System.out.println("|    THE SYSTEM WILL EXIT NOW     |");
699             System.out.println("+---------------------------------+");
700 
701         }
702         }while(option != 'E');
703 
704 
705     }
706     public static void RegisterCustomerProcess(Customer[] customer,Dealership
        dealership){
707                 int j = 0;
708                 while(j < 1){
709                     Scanner CustomerRegist = new Scanner(System.in);
710 
711                     customer[dealership.getNumCustomer()] = new Customer();
712 
713                     System.out.println("+---------------------------------+");
714                     System.out.println("|      Fill you information       |");
715                     System.out.println("+---------------------------------+");
716                       System.out.print("| Full Name      : ");
717                     String name = CustomerRegist.nextLine();
718                       System.out.print("| NRIC [no - ]   : ");
719                     String ic = CustomerRegist.nextLine();
720                       System.out.print("| Set a Password : ");
721                     String password = CustomerRegist.nextLine();
722                       System.out.print("| Home Address   : ");
723                     String address = CustomerRegist.nextLine();
724                       System.out.print("| Phone Number   : ");
725                     String phone = CustomerRegist.nextLine();
726 
727                     name = name.toUpperCase();
728                     address = address.toUpperCase();
729                         customer[dealership.getNumCustomer()].setCustomerData(name,
                            address, ic, phone, password);
730 
731                     System.out.print(customer[dealership.getNumCustomer()]);
732                     System.out.println("\n+---------------------------------+");
733                     System.out.println("|   is the information correct ?  |");
734                     System.out.println("+    [Y] = Yes   ||   [N] = NO    +");
735                     System.out.println("+---------------------------------+");
736                       System.out.print("| Answer : ");
737                     String answer = CustomerRegist.nextLine();
738 
739 
```

```
740                     if(answer.toUpperCase().charAt(0) == 'Y'){
741
742                         dealership.addCustomer(customer[dealership.getNumCustomer()]);
743
744                         System.out.println(customer[dealership.getNumCustomer() - 1]);
745                         System.out.println(dealership.getNumCustomer());
746                         System.out.println(customer.length);
747                         j++;
748                     }
749                     else{
750                         System.out.println("| Please fill the the form again.");
751                     }
752         }
753       }
754        public static void addNewCar(Car[] cars, Dealership dealership) throws Exception {
755                System.out.println("+---------------------------------+");
756                System.out.println("|          ADD NEW CAR            |");
757                System.out.println("+---------------------------------+");
758                    Scanner adminCar = new Scanner(System.in);
759                    System.out.println("+---------------------------------+");
760                    System.out.print("| Car Brand : ");
761                    String carB = adminCar.nextLine();
762                    System.out.print("| Car Model : ");
763                    String carM = adminCar.nextLine();
764                    System.out.print("| Car Type  : ");
765                    String carT = adminCar.nextLine();
766                    System.out.print("| Car Price : RM");
767                    double carp = adminCar.nextDouble();
768
769                    cars[dealership.getNumCars()] = new Car(carB, carM, carT, carp);
770
771                     dealership.addCarToInventory(cars[dealership.getNumCars()]);
772                try{
773                    writeFile(cars, dealership);
774                }
775                catch(Exception ex){}
776
777                    System.out.println("+---------------------------------+");
778                    System.out.println("| Car Added Successfully.");
779                    System.out.println("+---------------------------------+");
780
781        }
782        public static void RemoveCar(Car[] cars, Dealership dealership) throws
           FileNotFoundException, Exception{
783            Scanner adminCar = new Scanner(System.in);
784            int error = 1;
785            while(error == 1){
786            if(dealership.getNumCars() == 0){
787            System.out.println("+---------------------------------+");
788            System.out.println("|       NO CARS IN INVENTORY      |");
789            System.out.println("+---------------------------------+");
790                break;
791            }
792
793            System.out.println("+---------------------------------+");
794            System.out.println("|         REMOVE A CAR            |");
795            System.out.println("+---------------------------------+");
796            dealership.PrintCarList(cars);
797            System.out.println("+---------------------------------+");
798            System.out.println("|      Choose car to remove       |");
799            System.out.println("+---------------------------------+");
800            System.out.print("|  car number : ");
801            int indextoRemove = adminCar.nextInt();
802            indextoRemove = indextoRemove - 1;
803
804            if(cars[indextoRemove] == null){
805            System.out.println("+---------------------------------+");
806            System.out.println("|   A     CAR DOES NOT EXIST       |");
807            System.out.println("+---------------------------------+");
```

```
808          }
809
810          else {
811          Car carToRemove = cars[indextoRemove];
812          dealership.removeCarFromInventory(carToRemove, indextoRemove);
813
814          System.out.println("+---------------------------------+");
815          System.out.println("|     CAR REMOVED SUCCESSFULLY    |");
816          System.out.println("+---------------------------------+");
817          error = 0;
818
819          deleteCarData(cars, dealership, indextoRemove);
820          }
821
822          }
823
824      }
825      public static void importCar(Car[] cars, Dealership dealership,String Filename)
         throws Exception {
826
827          File CarImport = new File(Filename);
828          FileReader CarReader = new FileReader(CarImport);
829          BufferedReader bufferRead = new BufferedReader(CarReader);
830
831
832          int carnum = 0;
833              String line;
834              while((line = bufferRead.readLine()) != null){
835                  StringTokenizer tokenizer = new StringTokenizer(line,";");
836
837                  if(tokenizer.countTokens() == 4){
838                      carnum++;
839                      String Brand = tokenizer.nextToken();
840                      String Model = tokenizer.nextToken();
841                      String Type = tokenizer.nextToken();
842                      double price = Double.parseDouble(tokenizer.nextToken());
843
844                      cars[dealership.getNumCars()] = new Car(Brand, Model, Type, price);
845
846                      dealership.addCarToInventory(cars[dealership.getNumCars()]);
847
848                      try{
849                          writeFile(cars, dealership);
850                      }catch(Exception ex){}
851                  }
852              }
853                          System.out.println("| "+carnum+" Car(s) Imported.");
854                          System.out.println("| Car Imported Successfully.");
855                          bufferRead.close();
856
857              clearFile(Filename);
858      }
859      public static void writeFile(Car[] cars, Dealership dealership) throws Exception {
860          try{
861                          // EXPORT CAR
862                          File ExportCar = new File(CAR_FILE);
863                          FileWriter WriteCar = new FileWriter(ExportCar);
864                          PrintWriter PrintCar = new PrintWriter(WriteCar);
865                          for(int i =0;i<cars.length;i++){
866                              if(cars[i] != null){
867                                  Car[] carData = dealership.getCarInvetory();
868                                  PrintCar.println(carData[i].WriteString());
869                              }
870                          }
871                          PrintCar.close();
872          }
873          catch(Exception ex){}
874      }
875      public static void readFile(Car[] cars, Dealership dealership) throws Exception {
```

```
876           try{
877               File CarImport = new File(CAR_FILE);
878               FileReader CarReader = new FileReader(CarImport);
879               BufferedReader bufferRead = new BufferedReader(CarReader);
880
881               int carnum = 0;
882                           String line;
883                           while((line = bufferRead.readLine()) != null){
884                               StringTokenizer tokenizer = new StringTokenizer(line,";");
885
886                               if(tokenizer.countTokens() == 4){
887                                   carnum++;
888                                   String Brand = tokenizer.nextToken();
889                                   String Model = tokenizer.nextToken();
890                                   String Type = tokenizer.nextToken();
891                                   double price = Double.parseDouble(tokenizer.nextToken());
892                                   cars[dealership.getNumCars()] = new Car(Brand, Model,
                                      Type, price);
893
                                      dealership.addCarToInventory(cars[dealership.getNumCars()
                                      ]);
894                               }
895                           }
896                           bufferRead.close();
897           }
898           catch(Exception ex){}
899       }
900       public static void clearFile(String fileName){
901
902           try {
903               // CLEAR OUT IMPORt.TXT
904               FileWriter fileWriter = new FileWriter(fileName);
905               fileWriter.close();
906
907           } catch (IOException e) {
908               System.out.println("An error occurred: " + e.getMessage());
909       }
910
911       }
912       public static void deleteCarData(Car[] cars, Dealership dealership,int
          indextoRemove) throws FileNotFoundException, Exception {
913
914           File OriginalFile = new File(CAR_FILE);
915           File tempFile = new File("tempt.txt");
916
917               try {
918
919                   BufferedReader reader = new BufferedReader(new FileReader(OriginalFile));
920                   BufferedWriter writer = new BufferedWriter(new FileWriter(tempFile));
921
922                   String currentLine;
923                   int CurrentLineNumber = 1;
924                   int LineToDelete = (indextoRemove+1);
925
926                   while((currentLine = reader.readLine()) != null){
927                       if(CurrentLineNumber != LineToDelete){
928                           writer.write(currentLine);
929                           writer.newLine();
930                       }
931                       CurrentLineNumber++;
932                   }
933
934                    writer.close();
935                   reader.close();
936
937                   boolean successful = OriginalFile.delete();
938                   boolean successful2 = tempFile.renameTo(OriginalFile);
939                   System.out.println(successful);
940                   System.out.println(successful2);
```

```
941
942                 } catch (IOException e) {
943                     e.printStackTrace();
944                 }
945         }
946     }
947
```