**LAB # 09**

1. **Task # 01: Write a program and create the objects of classes in class car to explain the concept of composition. Create several classes as engine, doors, capacity and wheel having their individual methods attributes. The object of these classes are created in a car class and they are set as public. The object of this car class is created in Main method and this with the help of this object we can call other classes as well and can use their functionalities and design UML class diagram.**

**Solution:**

**public static void main(String[] args) {**

Car car1 = new Car();

// Access the engine,wheel and door class method through car obj

car1.getEn().engineWork();

car1.getWh().wheelCond("Stop");

car1.getDr().doorcond("open");

// car information

System.out.println("\n\n------- Car Information --------\n");

System.out.println(car1); }

**public class Car {**

private Engine en;

private Door dr;

private Wheel wh;

private Capacity ca;

public Car() {

en = new Engine(232435, "honda", "1300cc");

dr = new Door("black", "front left");

wh = new Wheel("tiger");

ca = new Capacity("1.6ghz", "40 KM");}

@Override

public String toString() { return en + "\n" + dr + "\n" + wh + "\n" + ca;}

public Engine getEn() {

return en;

}

public Door getDr() {

return dr;

}

public Wheel getWh() {

return wh;

}

public Capacity getCa() {

return ca;

}}

**public class Engine {**

int enno;

String manufacturer;

String cc;

public Engine(int enno, String manufacturer, String cc) {

this.enno = enno;

this.manufacturer = manufacturer;

this.cc = cc;

}

public void engineWork() {

System.out.println("Engine working....");

}

@Override

public String toString() {

return "Engine{" + "enno=" + enno + ", manufacturer=" + manufacturer + ", cc=" + cc + '}';

}}

**public class Door {**

String colour;

String which;

public Door(String colour, String which) {

this.colour = colour;

this.which = which;

}

void doorcond(String cond) {

System.out.println("DOor " + cond + " .....");

}

@Override

public String toString() {

return "Door{" + "colour=" + colour + ", which=" + which + '}';

}}

public class Wheel {

String company;

public Wheel(String company) {

this.company = company;

}

void wheelCond(String cond) {

System.out.println("Wheel " + cond + " .......");

}

@Override

public String toString() {

return "Wheel{" + "company=" + company + '}';

}}

**public class Capacity {**

String power;

String average;

public Capacity(String power, String average) {

this.power = power;

this.average = average;

}

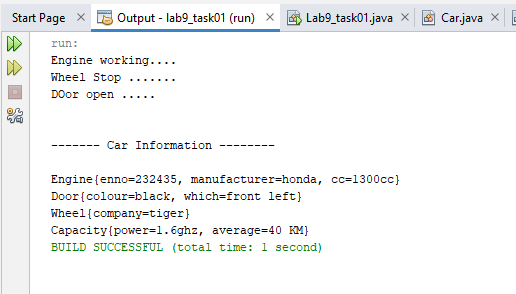
@Override

public String toString() {

return "Capacity{" + "power=" + power + ", average=" + average + '}';

}}

**Output:**



1. **Task # 02:** Write complete program for Flight's class, Time's class and Passenger's class with the concept of association and aggregation and design UML class diagram. Functions information also been given in the table.

**Solution:**

**public static void main(String[] args) {**

// TODO code application logic here

Scanner sc = new Scanner(System.in);

passenger p1 = new passenger("Ahsan", 79315, "Karachi", "Faislabad", "F231");

passenger p2 = new passenger("ali", 795615, "Karachi", "Faislabad", "F231");

passenger p3 = new passenger("ahmed", 432526, "Karachi", "Islamabad", "F231");

passenger p4 = new passenger("asad", 43624, "Karachi", "Islamabad", "F231");

passenger.addpassenger(p1);

passenger.addpassenger(p2);

passenger.addpassenger(p3);

passenger.addpassenger(p4);

List<passenger> pklist = new ArrayList<passenger>();

pklist.add(p1);

pklist.add(p2);

List<passenger> pklist1 = new ArrayList<passenger>();

pklist1.add(p3);

pklist1.add(p4);

flight f1 = new flight(p1.flightno, "Faislabad ", "08:00", "09:15", pklist);

flight f2 = new flight(p1.flightno, "Islamabad ", "08:00", "09:15", pklist1);

flight.addflight(f1);

flight.addflight(f2);

int res;

System.out.println("Please Select");

System.out.println(" 1 ) Passengers List");

System.out.println(" 2 ) Flight List");

System.out.print("Enter : ");

res = sc.nextInt();

sc.nextLine();

if (res == 1) {

passenger.printlist();

} else if (res == 2) {

f1.flightlist();

}

time t1 = new time();

System.out.println("Hour : " + t1.getHour() + " Minute : " + t1.getMinute());

}

**class passenger {**

String name;

int CNIC;

String from;

String to;

String flightno;

public passenger(String name, int CNIC, String from, String to, String flightno) {

this.name = name;

this.CNIC = CNIC;

this.from = from;

this.to = to;

this.flightno = flightno;

}

static List<passenger> passengers = new ArrayList<passenger>();

static void addpassenger(passenger p) {

passengers.add(p);

}

static void printlist() {

int i = 1;

System.out.println("---------------------PASSENGER DETAILS----------------");

for (passenger p : passengers) {

System.out.println(i + " )" + "\nName= " + p.name + "\nCNIC= " + p.CNIC + "\nfrom= " + p.from + "\nTo= " + p.to + "\nFlight no= " + p.flightno + "\n-----------------------------------------------------------");

i++;}}}

**class flight {**

String Flightno;

String Destination;

String departure;

String arrival;

int numberofpassengers;

List<passenger> passengers;

public flight(String Flightno, String Destination, String departure, String arrival, List<passenger> passengers) {

this.Flightno = Flightno;

this.Destination = Destination;

this.departure = departure;

this.arrival = arrival;

this.passengers = passengers;

for (passenger p : passengers) {

this.numberofpassengers++;

}}

static List<flight> flights = new ArrayList<flight>();

static void addflight(flight f) {

flights.add(f);

}

void flightlist() {

int i = 1;

for (flight f : flights) {

System.out.println(i + " ) " + "Flightno= " + Flightno + "\nDestination= " + Destination + "\ndeparture= " + departure + "\narrival=" + arrival + "\nnumberofpassengers= " + numberofpassengers + "\n---------------------------------------------------------------------------");

i++;}}}

**public class time {**

int hour;

int minute;

public time() {

LocalTime dt = LocalTime.now();

setHour(dt.getHour());

setMinute(dt.getMinute());

}

public void setHour(int hour) {

LocalTime dt = LocalTime.now();

this.hour = dt.getHour();

}

public void setMinute(int minute) {

LocalTime dt = LocalTime.now();

this.minute = dt.getMinute();

}

public int getHour() {

return hour;

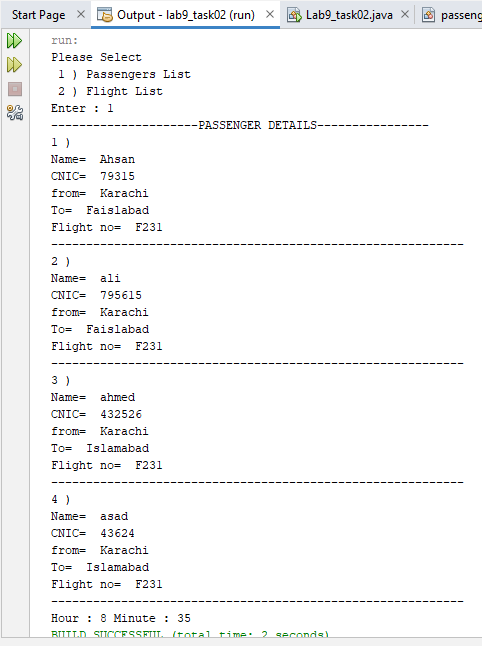
}

public int getMinute() {

return minute;

} }

**Output:**



**Task # 03: A company manages many stores. Each Store contains many Products. Implement Product, Store and Company classes using association and aggrigation concepts and design UML class diagram.**

**Solution:**

**public static void main(String[] args) {**

Product p1 = new Product("LAYS", " INDIA", 3212);

Product p2 = new Product("kurkure", " pakistan", 9812);

List<Product> products=new ArrayList<Product>();

products.add(p1);

products.add(p2);

Store s1 = new Store("ITZ\_STore", 1212, products);

Store s2 = new Store("ABC\_STore", 3612,products );

List<Store> stores=new ArrayList<Store>();

stores.add(s1);

stores.add(s2);

Company c1 = new Company("RTX\_COMPANY", "981ty", stores);

System.out.println("-------------------------------------------");

System.out.println(c1);

System.out.println("-------------------------------------------\n");

for (Store store : stores) {

System.out.println(store+"\n");

for (Product product : products) {

System.out.println(product);

}

System.out.println("\n");

}

c1.totalstores();

s1.totalproducts();} }

**public class Company {**

String name;

String int\_no;

Store st;

public Company(String name, String int\_no, Store st) {

this.name = name;

this.int\_no = int\_no;

this.st = st;

}

void totalstores(){

int total=0;

for (Store store : stores) {

total++;

}

System.out.println("TOtal STores: " +total);

}

@Override

public String toString() {

return "Company{" + "name=" + name + ", int\_no=" + int\_no + '}';

}}

**public class Store {**

String name;

int no;

Product pro;

public Store(String name, int no, Product pro) {

this.name = name;

this.no = no;

this.pro = pro;

}

void totalproducts(){

int total=0;

for (Product product : products) {

total++;

}

System.out.println("Total PRODUCTS In Store: "+ total);

}

@Override

public String toString() {

return "Store{" + "name=" + name + ", no=" + no + '}';

}}

**public class Product {**

String name;

String manufacture;

int code;

public Product(String name, String manufacture, int code) {

this.name = name;

this.manufacture = manufacture;

this.code = code;

}

@Override

public String toString() {

return "Product{" + "name=" + name + ", manufacture=" + manufacture + ", code=" + code + '}';

}}

**Output:**

