Lab 09 Lab tasks by 21SW159

**01. Create a multilevel inheritance hierarchy with a parent interface vehicle containing methods accelerate and decelerate, private properties: color, weight, model, with their setter and getters. Then the sub interfaces Bus, Truck, Car further extending classes SchoolBus, HondaCar, VigoTruck, with their implementation respectively.**

Code:

interface vehicle{

public void accelerate();

public void decelerate();

private String color="2";

private String model="3";

private int weight=4;

public void setColor(String c);

public String getColor();

public void setModel(String m);

public String getColor();

public void setWeight(int w);

public int getColor();

}

class Bus implements vehicle {

private String color="2";

private String model="3";

private int weight=4;

public void accelerate(){

System.out.println("accelerate");

}

public void accelerate(){

System.out.println("decelerate");

}

public void setColor(String c){

color = c;

}

public String getColor(){

return color;

}

public void setModel(String m){

model = m;

}

public String getModel(){

return model;

}

public void setWeight(int w){

weight = w;

}

public int getWeight(){

return weight;

}

}

class Car implements vehicle {

private String color="2";

private String model="3";

private int weight=4;

public void accelerate(){

System.out.println("accelerate");

}

public void accelerate(){

System.out.println("decelerate");

}

public void setColor(String c){

color = c;

}

public String getColor(){

return color;

}

public void setModel(String m){

model = m;

}

public String getModel(){

return model;

}

public void setWeight(int w){

weight = w;

}

public String getWeight(){

return weight;

}

}

class Truck implements vehicle {

private String color="2";

private String model="3";

private int weight=4;

public void accelerate(){

System.out.println("accelerate");

}

public void accelerate(){

System.out.println("decelerate");

}

public void setColor(String c){

color = c;

}

public String getColor(){

return color;

}

public void setModel(String m){

model = m;

}

public String getModel(){

return model;

}

public void setWeight(int w){

weight = w;

}

public String getWeight(){

return weight;

}

}

class SchoolBus extends Bus{

}

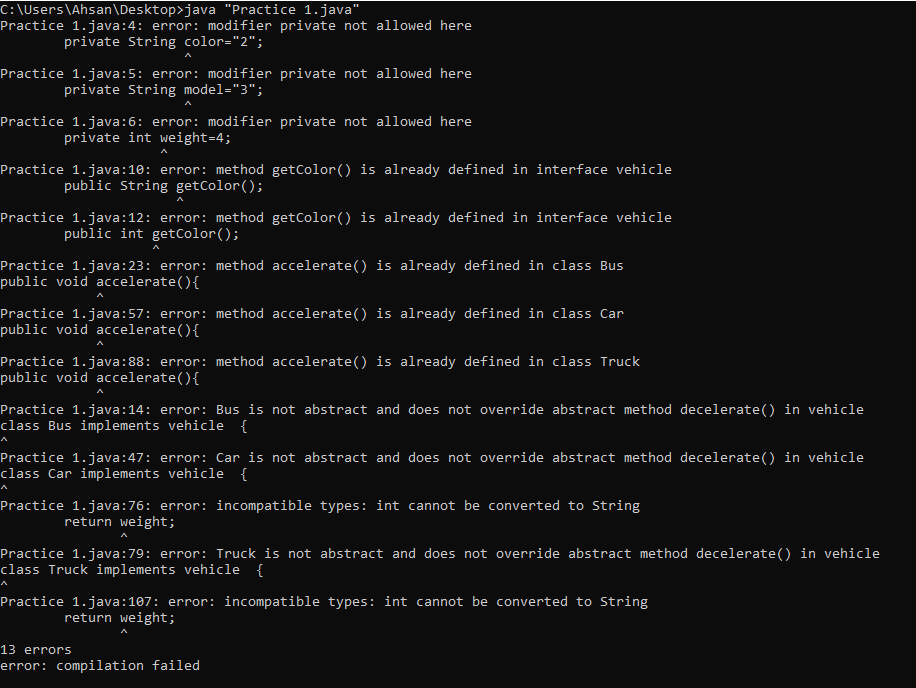
class HondaCar extends Car{

}

class VigoTruck extends Truck{

}

Output:



**02. Write a program with a mother class and an inherited daugther class. Both of them should have a method void display () that prints a message (different for mother and daugther). In the main define a daughter and call the display() method on it.**

**Code:**

class Demo{

public static void main(String[] args) {

Daughter d1 = new Daughter();

d1.display();

}

}

class Mother{

void display(){

System.out.println("Hello");

}

}

class Daughter extends Mother{

void display(){

System.out.println("Hi");

}

}

**Output:**



**03. Demonstrate the use of the following Keywords.**

**i. Super**

**ii. This**

**iii. Final**

**iv. Extends**

**v. Implements**

**Code:**

public class Main {

//The this keyword refers to the current object in a method or constructor.

int c;

public Main(int c){

this.c = c;

}

public static void main(String args[]) {

Animal Mycat = new Cat();

Mycat.animalSound();

Main my = new Main(3);

System.out.print(my.c);

}

}

class Animal {

public void animalSound() {

System.out.println("The animal makes a sound");

}

}

//extends is use to extend classes

class Cat extends Animal {

public void animalSound() {

super.animalSound();

System.out.println("The Cat says: meow maow");

}

}

interface fire {

public void Burn();

final int w = 35; //it is not constant value cannot be changed even if we try

}

//implementation (using implement keyword)

class Elementals implements fire{

public void Burn(){

System.out.println("Burning the jungle");

}

}

**Output:**

