SUBJECT: OBJECT ORIENTED PROGRAMMING

Submitted To: Engr. Asmatullah

Submitted By: Asadullah Samo (21SW036)

Dated: 04-09-2022

Lab: Lab-09 Tasks

**Question 01:**

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

Source Code:

**package** Lab\_09\_Tasks;

**import** java.util.Scanner;

**interface** Vehicle{

**public** **void** accelerate(**int** accelerate);

**public** **void** decelerate(**int** decelerate);

}

**interface** Car **extends** Vehicle{

// public void accelerate(int accelerate);

// public void decelerate(int decelerate);

}

**interface** Bus **extends** Vehicle{

// public void accelerate(int accelerate);

// public void decelerate(int decelerate);

}

**interface** Truck **extends** Vehicle{

// public void accelerate(int accelerate);

// public void decelerate(int decelerate);

}

**class** small\_Vehicle **implements** Vehicle{

**int** accelerate, decelarate;

**private** String color;

**private** **long** engine\_no;

**private** **float** weight;

**public** **void** setColor(String color) {

**this**.color = color;

}

**public** **void** setWeight(**float** weight) {

**this**.weight = weight;

}

**public** **void** setEngine\_no(**long** engine\_no) {

**this**.engine\_no = engine\_no;

}

**public** String getColor() {

**return** color;

}

**public** **float** getWeight() {

**return** weight;

}

**public** **long** getEngine\_no() {

**return** engine\_no;

}

@Override

**public** **void** accelerate(**int** accelerate){

**this**.accelerate = accelerate;

}

@Override

**public** **void** decelerate(**int** decelarate){

**this**.decelarate = decelarate;

}

} // end of class Honda

**class** Honda **extends** small\_Vehicle{

}

**class** VigoTuck **extends** small\_Vehicle{

}

**class** SchoolBus **extends** small\_Vehicle{

}

**public** **class** Lab\_09\_Task\_01 {

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

Honda h = **new** Honda();

Scanner sc = **new** Scanner(System.in);

// System.out.print("Enter the value for accelerate: ");

// int accelerate = sc.nextInt();

// System.out.print("Enter the value for decelerate: ");

// int decelerate = sc.nextInt();

System.out.print("Enter the color of Car: ");

String color = sc.nextLine();

h.setColor(color);

System.out.println("Enter the Car's weight: ");

**float** weight = sc.nextFloat();

h.setWeight(weight);

System.out.println("Enter the engine no: of Car");

**long** engine\_no = sc.nextLong();

h.setEngine\_no(engine\_no);

System.out.println("The color of Car is "+h.getColor());

System.out.println("The weight of Car is "+h.getWeight());

System.out.println("The engine no of Car is "+h.getEngine\_no());

SchoolBus sb = **new** SchoolBus();

System.out.print("Enter color for Bus: ");

String color2 = sc.next();

sb.setColor(color2);

System.out.println("The color of Raja Kamran is "+sb.getColor());

VigoTuck vt = **new** VigoTuck();

**float** weight2 = sc.nextFloat();

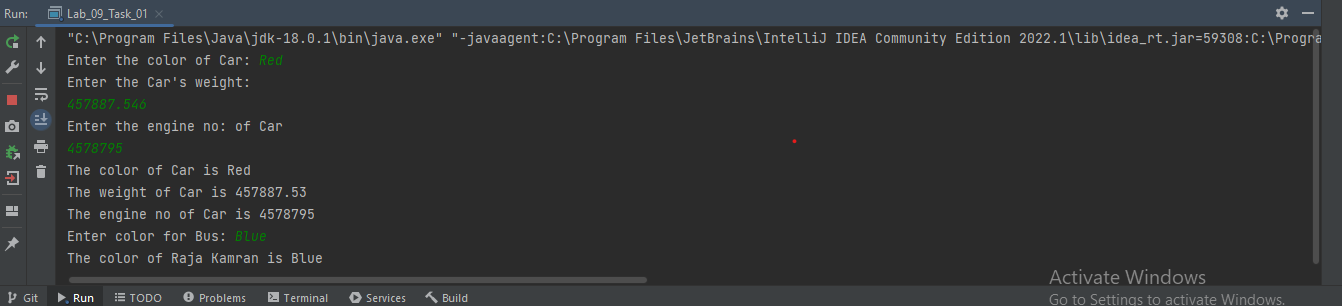
vt.setWeight(weight2);

System.out.println("The weight of vigo truck is "+vt.getWeight());

} // end of main() method

} // end of program

OUTPUT:



**Question 02:**

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

Source Code:

**package** Lab\_09\_Tasks;

**class** Mother{

**public** **void** display(){

System.out.println("This is mother class");

}

}

**class** Daughter **extends** Mother{

**public** **void** greet(){

System.out.println("Greet");

}

@Override

**public** **void** display(){

System.out.println("This is Daughter class");

}

}

**public** **class** Lab\_09\_Task\_02 {

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

Daughter d= **new** Daughter();

d.display();

Mother m = **new** Daughter(); // Object of Daughter is created with mother of reference

m.display();

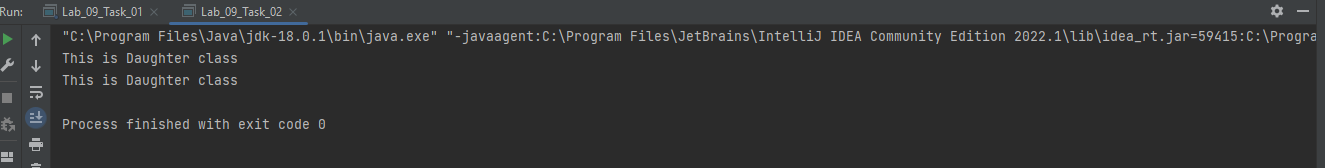
// m.greet() --> Not allowed (only those methods can be called which are present in Parent class)

// Daughter d1 = new Mother(); --> Not allowed

}

}

OUTPUT:

****

**Question 03:**

Demonstrate the use of the following keywords:

1. Super
2. This
3. Final
4. Extends
5. implements

Source Code:

**package** Lab\_09\_Tasks;

**class** Parent{

**public** Parent(**int** n, **char** s) {

**for**(**int** i=1; i<=n; i++){

System.out.print(s+" ");

}

System.out.println();

} // end of constructor

**private** **int** length, width;

**public** **void** setValues(**int** length, **int** width){

**this**.length = length; // Put private length attribute of this method equal to value of length that is passed to this method as parameter

**this**.width = width; // Put private width attribute of this method equal to value of width that is passed to this method as parameter

}

**public** **float** getValues(){

**return** length\*width;

}

**public** **void** display(){

System.out.println("I will be called before child constructor");

}

**public** **void** demonstrate(){

System.out.println("I am called by the object of child class");

}

} // end of class Parent

**class** Child **extends** Parent{

Child(){

**super**(7, '\*'); // calls the Constructor of Parent class

**super**.display(); // calls the method of parent class

}

}

// Demonstrating final keyword

**final** **class** Super{

**public** **void** show(){

System.out.println("My class will not be inherited");

}

}

//class Sub extends Super{ // This will give error (Cannot inherit from final class Super)

//

//}

**interface** Polygon{

**public** **void** area(**int** l, **int** b);

}

**class** Rectangle **implements** Polygon{

@Override

**public** **void** area(**int** l, **int** b){

System.out.println("The area of rectangle is "+l\*b);

}

}

**public** **class** Lab\_09\_Task\_03 {

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

// Demonstrating the use of super keyword

Child ch = **new** Child();

// Demonstrating the use of this keyword

ch.setValues(8, 9);

System.out.println("The area of rectangle is "+ch.getValues());

// Demonstrating final keyword

Super sp = **new** Super();

sp.show();

**final** **int** a = 5;

System.out.println("The value of a is "+a);

// a = 9; --> throws error (can't assign a value to final variable)

// 3rd use of final is final method can be overridden by child class

// Demonstrating the use of extends

ch.demonstrate(); // We accessed the method of Parent class using the object of child class

// Demonstrating the use of implements

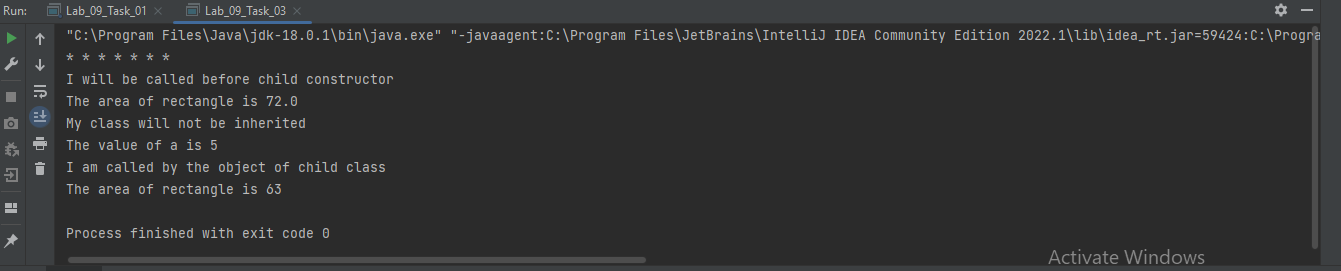
Rectangle r = **new** Rectangle();

r.area(7, 9); // Now rectangle implement the polygon method according to his need

} // end of main() method

} // end of program

OUTPUT:

****