**Group Activity 2 - Object and classes**

**PARENT CLASS - VEHICLE**

public class Vehicle {

private String name = "";

private String color = "";

private String model = "";

private String company = "";

private String engine = "800cc";

public static int count = 0; ***// increments every time an object is created***

public Vehicle() { ***// default constructor***

count++;

}

public Vehicle(String name, String color, String model, String company) {

this.name = name;

this.color = color;

this.model = model;

this.company = company;

count++;

}

public Vehicle(String name, String color, String model, String company, String engine) {

this.name = name;

this.color = color;

this.model = model;

this.company = company;

this.engine = engine;

count++;

}

public void setName(String name) {

this.name = name;

}

public void setColor(String color) {

this.color = color;

}

public void setModel(String model) {

this.model = model;

}

public void setCompany(String company) {

this.company = company;

}

public void setEngine(String engine) {

this.engine = engine;

}

public String getName() {

return this.name;

}

public String getColor() {

return this.color;

}

public String getModel() {

return this.model;

}

public String getCompany() {

return this.company;

}

private String getEngine() {

return this.engine;

}

public String getTopSpeed() {

String a = getEngine();

if (a == "800cc") {

return "100 km/h";

} else {

return "120 km/h";

}

}

public static String getVehicle() {

return "you have given the name and color of the vehicle";

}

public String getInfo() { ***// to illustrate polymorphism and method overriding***

return "this is a vehicle";

}

}

**SUBCLASS-1: CAR**

public class Car extends Vehicle {

***// default value***

private boolean powerSteering = false;

private boolean ledScreen = true;

public Car(String name, String color, String model, String company, String engine, boolean powerSteering,

boolean ledScreen) {

super(name, color, model, company, engine);

this.powerSteering = powerSteering;

this.ledScreen = ledScreen;

}

public String getColor() {

return "The color of your car is " + super.getColor();

}

public void setPowerSteering(boolean powerSteering) {

this.powerSteering = powerSteering;

}

public void setLedScreen(boolean ledScreen) {

this.ledScreen = ledScreen;

}

public boolean getPowerSteering() {

return this.powerSteering;

}

public boolean getLedScreen() {

return this.ledScreen;

}

***// to illustrate polymorphism and method overriding***

public String getInfo() {

return "this is a car";

}

}

**SUBCLASS-2: BIKE**

public class bike extends Vehicle {

private boolean fourStroke = false;

public bike(String name, String color, String model, String company, String engine, boolean fourStroke) {

super(name, color, model, company, engine);

this.fourStroke = fourStroke;

}

public void setFourStroke(boolean fourStroke) {

this.fourStroke = fourStroke;

}

public boolean getFourStroke(boolean fourStroke) {

return this.fourStroke;

}

**// to illustrate polymorphism and method overriding**

public String getInfo() {

return "this is a bike";

}

}

**MAIN METHOD**

public class Main {

public static void main(String[] args) {

Vehicle Vehicle1 = new Vehicle();

System.out.println(Vehicle1.getColor());

Vehicle1.setName("amaze");

System.out.println(Vehicle1.getName());

System.out.println(Vehicle1.count);

Vehicle Vehicle2 = new Vehicle("civic", "black", "2012", "honda");

System.out.println(Vehicle2.getColor());

Vehicle2.setColor("white");

System.out.println(Vehicle2.count);

System.out.println(Vehicle2.getColor());

System.out.println(Vehicle2.getCompany());

Vehicle Vehicle3 = new Vehicle("seltos", "red", "2020", "kia", "1200cc");

System.out.println(Vehicle3.getTopSpeed());

System.out.println(Vehicle3.count);

***// calling static functions does not requires to create an Object***

System.out.println(Vehicle.getVehicle());

Car Car1 = new Car("sonet", "white", "2019", "kia", "800cc", true, false);

System.out.println(Car1.getPowerSteering());

System.out.println(Car1.getColor());

System.out.println(Vehicle1.getInfo());***// polymorphism***

Vehicle Vehicle4 = new Car("sonet", "white", "2019", "kia", "800cc", true, false);

System.out.println(Vehicle4.getInfo());***// polymorphism***

Vehicle Vehicle5 = new bike("pulsar", "green", "2010", "yamaha", "600cc", true);

System.out.println(Vehicle5.getInfo());***// polymorphism***

}

}

**EXPLANATION**

Overview:

In the above example, **Vehicle** is our **Parent** **/ Base** Class. The **Child / Subclass** **Car** and **Bike** inherits properties from Vehicle Class. The various concepts and properties used are mentioned in the code through comments.

Concepts used in the above Example:

* Parent and Child Class
* Method Overloading and Method Overriding
* Polymorphism
* The behaviour of Contractors in different scenarios