Selenium WebDriver Training

Java OOPs - Polymorphism

Testleaf
Always Ahead

Polymorphism

Poly – Many

Morphism- Forms

Father

Child

Employee

Polymorphism

- The ability of a class to provide different implementations of a method

- Java allows us to perform the same action in many different ways

Polymorphism Types

- Static Polymorphism / Compile-time polymorphism/Early Binding
- Dynamic Polymorphism/ Run-time Polymorphism/ Late Binding

Static Polymorphism (Overloading)

- When 2 or more methods inside the same class
- Have same signature except the input arguments
    a) Number of argument is different
    b) Type of argument is different
- Purpose
    Simplicity

Static Polymorphism

- Works within the same class and with same method name having different argument counts / different types of arguments

Example code

Calculator add functions with different inputs

normal methods with same method name with different arguments

```
add(int x, int y);
add(float a, float b);
add(int x, int y, int z)
```

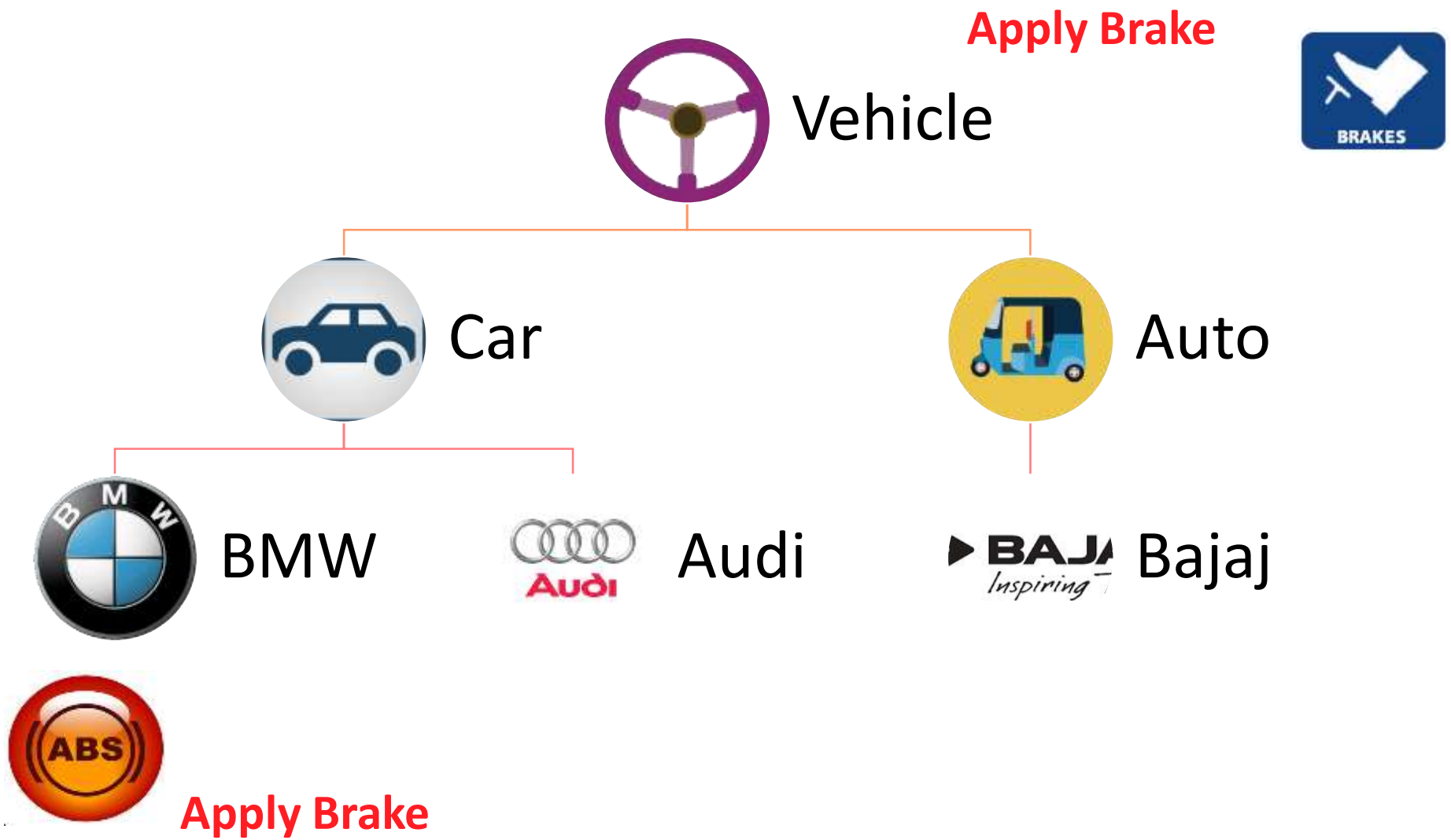
Dynamic Polymorphism

- Dynamic polymorphism can be achieved by Method Overriding
- Method Overriding -> different ways a method can behave based on IS –A relationship

Dynamic Polymorphism

- Method Overriding works based on inheritance.
- It allows a subclass to provide its own implementation of a method that is already defined in its superclass

Method Overriding



Classroom : (Breakout)

Method Overloading

Create class Calculator with below methods:

- **2 methods for add**.
  1 method with 2 int args,
  1 method with 3 int args
- **2 methods for multiple.**
  1 method with 2 double args,
  1 method with 2 float args,

Create Object for calculator class and call all the methods

Method Overriding:

Create a two classes AndroidPhone and SmartPhone

Create the same method (takeVideo) in AndroidPhone and SmartPhone (SmartPhone extends Android Phone)

Execute the method using child class object

Summary

- Polymorphism – **Method Overloading & Method Overriding**
- Method Overloading –Same class with same method name with different arguments
- Method Overriding –Same method name , same argument in different classes
- Method Overriding –keyword **extends**