Object-Oriented Code Complexity Matrix

Group Details:

Group Number: 09

Batch: Third Year (First Semester - Weekend)

|  |  |  |
| --- | --- | --- |
|  | Student Registration Number | Student Name |
| 1 | IT18050318 | M. A. Zeid |
| 2 | IT17029896 | D.S. Jiffry |
| 3 | IT18060690 | M.R.M. Rifan |
| 4 | IT18200034 | M.A.F.Hasna |

Contents

[1.0 Introduction 3](#_Toc40047886)

[2.0 Proposal of new complexity matrix 4](#_Toc40047887)

[3.0 Factors Affecting the Complexity Matrix 5](#_Toc40047888)

[3.1 Coupling due to Inheritance 5](#_Toc40047889)

[3.2 Decoupling due to Polymorphism 5](#_Toc40047890)

[3.3 Coupling by Association or Aggregation 5](#_Toc40047891)

[3.4 Coupling due to Composition 5](#_Toc40047892)

[4.0 Formulating the Equations 6](#_Toc40047893)

[4.1 Main Variable Names and their Meaning 6](#_Toc40047894)

[4.2 Core Equation 6](#_Toc40047895)

[5.0 Factors of Proposed Measure 7](#_Toc40047896)

[5.1 Inheritance Factor 7](#_Toc40047897)

[5.2 Polymorphism Factor 7](#_Toc40047898)

[6.0 Calculation of Complexity of Executable Java Programs 9](#_Toc40047899)

[6.1 Program One 9](#_Toc40047900)

[6.2 Program Two 12](#_Toc40047901)

[6.3 Program Three 15](#_Toc40047902)

[6.4 Program Four 17](#_Toc40047903)

[7.0 Conclusion 21](#_Toc40047904)

[8.0 References 22](#_Toc40047905)

# Introduction

Object oriented programming paradigm has been the DeFacto standard in many software systems since its introduction to the programming space in the early 90’s. Object-oriented thinking helps modelling complex programs easily through concepts such as abstraction, inheritance, encapsulation etc. Since many successful object-oriented programming languages became popular (i.e. Java and C++) and many modeling tools became popular such as UML, it soon became an industry standard to write programs.

Even though many advantages were provided by Object oriented programming it required talented engineers to model them in such a way that those advantages are utilized, and since it is conceptually challenging to grasp, many systems were poorly designed resulting in many anti patterns and poorly performing and scalable systems. Complexity of Object-oriented systems increase exponentially with size of the system and introduce coupling between parts of the system making the system fragile which makes certain classes almost unmodifiable since changing that would break or change the entire programs behavior. That should not discourage the use of object-oriented programming because it fits perfectly into many use cases but as an organization it is critical to have a quantitative measure of the impact of each entity (I.e. a class) on the rest of the system so organizations can make informed decisions on maintaining code quality, refactoring or changing existing code and manage complexity of the system.

This paper reviews the solution for code complexity that proposes a new model for OO code complexity. The new model has analyzed into OOP complexity metrics of inheritance, polymorphism, abstraction and composition as factors. The proposed model is based on readability, understandability, maintainability, reusability, extensibility and consistency of the programming code.

# 2.0 Proposal of new complexity matrix

The purpose of this report is to introduce a new object-oriented code complexity matrix that allows an organization to measure the impact of each class on a closed system of finite number of classes. The calculated matrix values should reflect which classes are the most critical classes and give an idea how changing that would impact the rest of the system relative to each other. This matrix measures the impact of one class based on the main object-oriented relationships Inheritance, aggregation, composition and association on the rest of the system.

# 3.0 Factors Affecting the Complexity Matrix

## 3.1 Coupling due to Inheritance

Inheritance of one class's properties and methods can introduce a tight coupling between classes specially if the methods aren't overloaded or overridden by the child class. Due to inheritance alone deleting the parent class, changing variable names or method names may break the child classes. The second part is due to methods and functionality. If the child classes use the super class methods directly the functionality of child class may change if the parent class is changed, resulting the child classes to behave differently even if they remained unchanged.

## 3.2 Decoupling due to Polymorphism

Method overloading and method overriding will decouple this relationship between the child and parent class. For instance, if the child class has overloaded or overridden the parent class if the parent class changes the methods of the child class will not be affected by it.

## 3.3 Coupling by Association or Aggregation

Association may also result in coupling. If the methods are modified in the classes that provides the methods the classes that uses these methods may behave differently introducing a form of coupling.

Changing method names or parameter types may also break all associated classes.

## 3.4 Coupling due to Composition

Composition relationships introduce a very high amount of coupling. If the container class is deleted the contained classes would cease to exist. So, the number of composite classes may increase the criticality of the container class.

# 4.0 Formulating the Equations

## 4.1 Main Variable Names and their Meaning

|  |  |
| --- | --- |
| **Variable** | **Meaning** |
| **I**f | Inheritance factor |
| **P**f | Polymorphism factor |
| **A**f | Association factor |
| **C**f | Composition factor |

## 4.2 Core Equation

  Ci = If – Pf + Af + Cf

Class impact on the system

* Pf, Af, Cf - Increases the impact so they are added.
* If – Decreases the impact so it is decreased.

# 5.0 Factors of Proposed Measure

## 5.1 Inheritance Factor

To consider the complexity added considering the level of inheritance statements are given below:

1. You pick the target class
2. Scan rest of the classes
3. then when you see extends keyword with target class name then we put I for IC and Weight is 2
4. When you see the super keyword, you put 1 for IC and Weight is also 1

## 5.2 Polymorphism Factor

To consider the complexity added considering methods overloading and method overriding. Statements are given below:

1. Classes you have to see that extends the target class
2. Every time you see method name that is similar to a method in the class, then you put 1.

## 5.3 Composition

1. Can through the target class
2. Every time you find class keywork inside a class (Nested Class), then weight is 1

## 5.4 Association

1. Scan every class other than the target class
2. Every time you find the target class name inside the body of the other classes, then weight is 1.
3. Every time you find the method name that is in the target class inside the body of the other classes, then weight is 2.
4. The condition is to scan the classes which are not target class and class that extends target class.

# Calculation of Complexity of Executable Java Programs

## Factors and Its Meaning

|  |  |
| --- | --- |
| **Short Form** | **Meaning** |
| If | Inheritance Factor |
| wI | Weight due to Inheritance |
| nwI | Net weight due to Inheritance |
| Pf | Polymorphism Factor |
| wP | Weight due to Polymorphism |
| nwP | Net weight due to Polymorphism |
| Af | Association Factor |
| wA | Weight due to Association |
| nwA | Net weight due to Association |
| Cf | Composition Factor |
| wC | Weight due to Composition |
| nwC | Net weight due to Composition |

## Programs and Calculation of Complexity

There are four main programs. Each program has two kinds of calculation that checks two different complexities. Each program considers different classes as target class.

### 6.2.1 Program One

This program considers the Person as the target class.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Line No** | **Statement** | **Inheritance** | | | **Polymorphism** | | | **Association** | | | **Composition** | | | **Total** |
| **If** | **Wl** | **nwl** | **pf** | **wp** | **nwp** | **Af** | **wA** | **nwA** | **Cf** | **wC** | **nwC** |
| 1 | import java.util.ArrayList; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | import java.util.List; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | class Person { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | private String name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | private String email; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | static String owner = "John"; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | public Person(String name, String email) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | this.name = name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | this.email = email; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | public String getEmail() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | return email; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 18 | public void setEmail(String email) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 19 | this.email = email; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 21 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 22 | public String getName() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23 | return name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 24 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 26 | public void setName(String name) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 27 | this.name = name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 28 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 29 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 30 | // toString() method to print info of Person |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 32 | return ("Name is: "+getName() + "\n"  + "Email is:"+getEmail()); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 33 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 34 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 35 | static class StaticNestedClass { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 36 | void display() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 37 | System.out.println("owner is: " + owner); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 38 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 39 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 41 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 42 | // Inheritance |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 43 | class Guest extends Person { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 44 | private int roomNo; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 45 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 46 | public Guest(String name, String email, int roomNo) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 47 | super(name,email); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 48 | this.roomNo = roomNo; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 49 | this.setName(name); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 50 | this.setEmail(email); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 51 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 52 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 53 | public int getRoomNo() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 54 | return roomNo; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 55 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 56 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 57 | public void setRoomNo(int roomNo) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 58 | this.roomNo = roomNo; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 59 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 61 | public void setRoomPrice(double priceforNormal){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 62 | priceforNormal = 18000.00; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 63 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 64 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 65 | public void setRoomPrice(double price, double extraServiceCharge){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 66 | extraServiceCharge = 1000.00; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 67 | price = 42000.00 + extraServiceCharge; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 68 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 69 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 70 | // overriding toString() method |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 71 | @Override |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 72 | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 73 | return (super.toString() +  "\nRoom No is: "+getRoomNo()); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 74 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 75 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 76 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 77 | //Composition |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 78 | class Hotel { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 79 | private final List<Guest> guests; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 80 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 81 | public Hotel(List<Guest> guests) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 82 | this.guests = guests; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 83 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 84 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 85 | public List<Guest> getTotalBooksInLibrary() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 86 | return guests; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 87 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 88 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 89 | public void addGuest(Guest guest) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 90 | guests.add(guest); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 91 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 92 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 93 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 94 | public class Main { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 95 | public static void main(String[] args) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 96 | Guest g1 = new Guest(1111,"MR. Kent","kent@gmail.com"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 97 | Guest g2 = new Guest(2222,"MR. Sheen","sheen@gmail.com"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 98 | Guest g3 = new Guest(3333,"Ms. Jane","jane@gmail.com"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 99 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 100 | // Creating the list which contains the no.of books |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 101 | List<Guest> guests = new ArrayList<>(); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 102 | guests.add(g1); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 103 | guests.add(g2); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 104 | guests.add(g3); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 105 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 106 | // Composition: There can be no guests without a hotel. |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 107 | // Association: One Hotel can have many guests |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 108 | Hotel hotel = new Hotel(guests); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 109 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 110 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Total** | | | | | | | | | | | | | |  |

This program considers the Hotel as the target class.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Line No** | **Statement** | **Inheritance** | | | **Polymorphism** | | | **Association** | | | **Composition** | | | **Total** |
| **If** | **Wl** | **nwl** | **pf** | **wp** | **nwp** | **Af** | **wA** | **nwA** | **Cf** | **wC** | **nwC** |
| 1 | import java.util.ArrayList; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 | import java.util.List; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | class Person { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | private String name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | private String email; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | static String owner = "John"; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | public Person(String name, String email) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 | this.name = name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | this.email = email; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | public String getEmail() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | return email; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 18 | public void setEmail(String email) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 19 | this.email = email; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 21 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 22 | public String getName() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23 | return name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 24 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 25 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 26 | public void setName(String name) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 27 | this.name = name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 28 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 29 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 30 | // toString() method to print info of Person |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 32 | return ("Name is: "+getName() + "\n"  + "Email is:"+getEmail()); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 33 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 34 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 35 | static class StaticNestedClass { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 36 | void display() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 37 | System.out.println("owner is: " + owner); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 38 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 39 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 41 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 42 | // Inheritance |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 43 | class Guest extends Person { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 44 | private int roomNo; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 45 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 46 | public Guest(String name, String email, int roomNo) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 47 | super(name,email); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 48 | this.roomNo = roomNo; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 49 | this.setName(name); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 50 | this.setEmail(email); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 51 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 52 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 53 | public int getRoomNo() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 54 | return roomNo; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 55 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 56 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 57 | public void setRoomNo(int roomNo) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 58 | this.roomNo = roomNo; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 59 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 61 | public void setRoomPrice(double priceforNormal){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 62 | priceforNormal = 18000.00; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 63 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 64 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 65 | public void setRoomPrice(double price, double extraServiceCharge){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 66 | extraServiceCharge = 1000.00; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 67 | price = 42000.00 + extraServiceCharge; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 68 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 69 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 70 | // overriding toString() method |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 71 | @Override |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 72 | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 73 | return (super.toString() +  "\nRoom No is: "+getRoomNo()); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 74 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 75 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 76 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 77 | //Composition |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 78 | class Hotel { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 79 | private final List<Guest> guests; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 80 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 81 | public Hotel(List<Guest> guests) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 82 | this.guests = guests; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 83 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 84 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 85 | public List<Guest> getTotalBooksInLibrary() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 86 | return guests; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 87 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 88 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 89 | public void addGuest(Guest guest) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 90 | guests.add(guest); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 91 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 92 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 93 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 94 | public class Main { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 95 | public static void main(String[] args) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 96 | Guest g1 = new Guest(1111,"MR. Kent","kent@gmail.com"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 97 | Guest g2 = new Guest(2222,"MR. Sheen","sheen@gmail.com"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 98 | Guest g3 = new Guest(3333,"Ms. Jane","jane@gmail.com"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 99 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 100 | // Creating the list which contains the no.of books |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 101 | List<Guest> guests = new ArrayList<>(); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 102 | guests.add(g1); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 103 | guests.add(g2); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 104 | guests.add(g3); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 105 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 106 | // Composition: There can be no guests without a hotel. |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 107 | // Association: One Hotel can have many guests |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 108 | Hotel hotel = new Hotel(guests); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 109 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 110 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Total** | | | | | | | | | | | | | |  |

### 6.2.2 Program Two

This program considers the Bicycle as the target class.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Line No** | **Java Program** | **Inheritance** | | | **Polymorphism** | | | **Association** | | | **Composition** | | | **Total** |
| **If** | **Wl** | **nwl** | **pf** | **wp** | **nwp** | **Af** | **wA** | **nwA** | **Cf** | **wC** | **nwC** |
| 1 | import java.util.\*; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | // base class |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | class Bicycle { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | // the Bicycle class has two fields |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | public int bicycleId; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | public String bicycleName; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | public int gear; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | public int speed; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | // the Bicycle class has one constructor |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | public Bicycle(int id, String name, int gear, int speed) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 | this.bicycleId = id; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | this.bicycleName = name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | this.gear = gear; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 | this.speed = speed; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 19 | // the Bicycle class has three methods |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 | public void applyBrake(int decrement) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 21 | speed -= decrement; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 22 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 24 | public void speedUp(int increment) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 25 | speed += increment; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 26 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 27 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 28 | // toString() method to print info of Bicycle |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 29 | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 30 | return("No of gears are "+gear  +"\n"  + "speed of bicycle is "+speed); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 32 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 33 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 34 | // derived class |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 35 | class MountainBike extends Bicycle { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 36 | // the MountainBike subclass adds more field |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 37 | public int seatHeight; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 38 | public double newPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 39 | public double oldPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 | public String driverName; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 41 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 42 | // the MountainBike subclass has one constructor |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 43 | public MountainBike(int id, String name, int gear,int speed,int stHeight,String driName,double currentPrice, double previousPrice) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 44 | // invoking base-class(Bicycle) constructor |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 45 | super(id,name,gear,speed); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 46 | seatHeight = stHeight; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 47 | this.driverName = driName; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 48 | newPrice = currentPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 49 | oldPrice = previousPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 50 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 51 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 52 | // the MountainBike subclass adds one more method |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 53 | public void setHeight(int newValue) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 54 | seatHeight = newValue; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 55 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 56 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 57 | public void setPrice(double value) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 58 | newPrice = value; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 59 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 61 | //Method Overloading |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 62 | public void setPrice(double value, double additionalPay){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 63 | additionalPay = 10000.00; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 64 | oldPrice = value + additionalPay; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 65 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 66 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 67 | // overriding toString() method |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 68 | // of Bicycle to print more info |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 69 | @Override |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 70 | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 71 | return (super.toString() +  "\nseat height is "+seatHeight); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 72 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 73 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 74 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 75 | //composition |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 76 | class Showroom { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 77 | private final List<MountainBike> mountainbikes; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 78 | static String name = "Cycles Pvt Lmtd"; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 79 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 80 | Showroom (List<MountainBike> mountainbikes) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 81 | this.mountainbikes = mountainbikes; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 82 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 83 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 84 | public List<MountainBike> getTotalBikesInShowroom(){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 85 | return mountainbikes; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 86 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 87 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 88 | static class StaticNestedClass { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 89 | void display() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 90 | System.out.println("owner is: " + owner); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 91 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 92 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 93 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 94 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 95 | // driver class |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 96 | public class Test { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 97 | public static void main(String args[]) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 98 | MountainBike mb1 = new MountainBike( 1,"pulsar",3, 100,45,"John",1000000.00,150000.00); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 99 | MountainBike mb2 = new MountainBike( 1,"porche",3, 100,45,"kumar",1000000.00,150000.00); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 100 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 101 | List<MountainBike> mountainbikes = new ArrayList<MountainBike>(); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 102 | mountainbikes.add(mb1); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 103 | mountainbikes.add(mb2); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 104 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 105 | Showroom showroom = new Showroom(mountainbikes); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 106 | System.out.println(mb1.toString()); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 107 | System.out.println(mb1.driverName+" is a driver of Bicycle Id: "+mb1.bicycleId); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 108 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 109 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Total** | | | | | | | | | | | | | |  |

This program considers the Showroom as the target class.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Line No** | **Java Program** | **Inheritance** | | | **Polymorphism** | | | **Association** | | | **Composition** | | | **Total** |
| **If** | **Wl** | **nwl** | **pf** | **wp** | **nwp** | **Af** | **wA** | **nwA** | **Cf** | **wC** | **nwC** |
| 1 | import java.util.\*; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 3 | // base class |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 4 | class Bicycle { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 5 | // the Bicycle class has two fields |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 6 | public int bicycleId; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 7 | public String bicycleName; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 8 | public int gear; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 9 | public int speed; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 11 | // the Bicycle class has one constructor |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 12 | public Bicycle(int id, String name, int gear, int speed) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 13 | this.bicycleId = id; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14 | this.bicycleName = name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 15 | this.gear = gear; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 16 | this.speed = speed; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 19 | // the Bicycle class has three methods |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 20 | public void applyBrake(int decrement) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 21 | speed -= decrement; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 22 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 23 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 24 | public void speedUp(int increment) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 25 | speed += increment; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 26 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 27 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 28 | // toString() method to print info of Bicycle |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 29 | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 30 | return("No of gears are "+gear  +"\n"  + "speed of bicycle is "+speed); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 31 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 32 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 33 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 34 | // derived class |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 35 | class MountainBike extends Bicycle { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 36 | // the MountainBike subclass adds more field |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 37 | public int seatHeight; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 38 | public double newPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 39 | public double oldPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 40 | public String driverName; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 41 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 42 | // the MountainBike subclass has one constructor |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 43 | public MountainBike(int id, String name, int gear,int speed,int stHeight,String driName,double currentPrice, double previousPrice) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 44 | // invoking base-class(Bicycle) constructor |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 45 | super(id,name,gear,speed); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 46 | seatHeight = stHeight; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 47 | this.driverName = driName; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 48 | newPrice = currentPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 49 | oldPrice = previousPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 50 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 51 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 52 | // the MountainBike subclass adds one more method |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 53 | public void setHeight(int newValue) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 54 | seatHeight = newValue; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 55 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 56 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 57 | public void setPrice(double value) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 58 | newPrice = value; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 59 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 60 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 61 | //Method Overloading |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 62 | public void setPrice(double value, double additionalPay){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 63 | additionalPay = 10000.00; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 64 | oldPrice = value + additionalPay; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 65 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 66 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 67 | // overriding toString() method |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 68 | // of Bicycle to print more info |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 69 | @Override |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 70 | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 71 | return (super.toString() +  "\nseat height is "+seatHeight); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 72 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 73 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 74 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 75 | //composition |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 76 | class Showroom { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 77 | private final List<MountainBike> mountainbikes; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 78 | static String name = "Cycles Pvt Lmtd"; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 79 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 80 | Showroom (List<MountainBike> mountainbikes) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 81 | this.mountainbikes = mountainbikes; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 82 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 83 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 84 | public List<MountainBike> getTotalBikesInShowroom(){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 85 | return mountainbikes; |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 86 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 87 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 88 | static class StaticNestedClass { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 89 | void display() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 90 | System.out.println("owner is: " + owner); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 91 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 92 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 93 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 94 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 95 | // driver class |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 96 | public class Test { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 97 | public static void main(String args[]) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 98 | MountainBike mb1 = new MountainBike( 1,"pulsar",3, 100,45,"John",1000000.00,150000.00); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 99 | MountainBike mb2 = new MountainBike( 1,"porche",3, 100,45,"kumar",1000000.00,150000.00); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 100 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 101 | List<MountainBike> mountainbikes = new ArrayList<MountainBike>(); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 102 | mountainbikes.add(mb1); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 103 | mountainbikes.add(mb2); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 104 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 105 | Showroom showroom = new Showroom(mountainbikes); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 106 | System.out.println(mb1.toString()); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 107 | System.out.println(mb1.driverName+" is a driver of Bicycle Id: "+mb1.bicycleId); |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 108 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 109 | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Total** | | | | | | | | | | | | | |  |

### 6.3 Program Three

This program considers the Bank as the target class.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Line No** | **Java Program** | **Inheritance** | | | **Polymorphism** | | | **Association** | | | | **Composition** | | | **Total** |
| **If** | **Wl** | **nwl** | **pf** | **wp** | **nwp** | **Af** | **wA** | **nwA** | **Cf** | | **wC** | **nwC** |  |
| 1 | import java.io. \*; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2 | import java.util.\*; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 4 | // class bank |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5 | class Bank { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 6 | private String name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7 | private final List<Employee> employees; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 8 | static String location = "Colombo"; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 10 | Bank (String name, List<Employee> employees) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 11 | this.name = name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 12 | this.employees = employees; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 13 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 15 | public String getBankName() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 16 | return this.name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 17 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 18 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 19 | public List<Employee> getTotalEmployeesInBank(){ |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 20 | return employees; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 21 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 22 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 23 | static class StaticNestedClass { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 24 | void display() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 25 | System.out.println("owner is: " + owner); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 26 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 27 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 28 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 29 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 30 | class Employee { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 31 | private String name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 32 | private String address; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 33 | private int number; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 34 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 35 | public Employee(String name, String address, int number) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 36 | System.out.println("Constructing an Employee"); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 37 | this.name = name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 38 | this.address = address; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 39 | this.number = number; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 40 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 41 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 42 | public void mailCheck() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 43 | System.out.println("Mailing a check to " + this.name + " " + this.address); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 44 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 45 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 46 | public String toString() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 47 | return name + " " + address + " " + number; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 48 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 49 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 50 | public String getName() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 51 | return name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 52 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 53 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 54 | public String getAddress() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 55 | return address; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 56 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 57 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 58 | public void setAddress(String newAddress) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 59 | address = newAddress; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 60 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 61 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 62 | public int getNumber() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 63 | return number; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 64 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 65 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 66 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 67 | class Salary extends Employee { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 68 | private double salary; // Annual salary |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 69 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 70 | public Salary(String name, String address, int number, double salary) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 71 | super(name, address, number); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 72 | setSalary(salary); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 73 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 74 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 75 | public void mailCheck() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 76 | System.out.println("Within mailCheck of Salary class "); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 77 | System.out.println("Mailing check to " + getName()  + " with salary " + salary); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 78 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 79 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 80 | public double getSalary() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 81 | return salary; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 82 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 83 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 84 | public void setSalary(double newSalary) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 85 | if(newSalary >= 0.0) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 86 | salary = newSalary; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 87 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 88 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 89 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 90 | //Method overloading |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 91 | public void setSalary(double newSalary, int months) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 92 | months = 12; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 93 | System.out.println(" Annual salary pay for: " + getName()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 94 | salary = newSalary \* 12; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 95 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 96 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 97 | public double computePay() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 98 | System.out.println("Computing salary pay for " + getName()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 99 | return salary/52; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 100 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 101 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 102 | //Method overloading |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 103 | public double computePay(double newSalary, double ETF) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 104 | ETF = 1000.00; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 105 | System.out.println("Computing salary pay for " + getNumber()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 106 | newSalary = salary - ETF; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 107 | return newSalary; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 108 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 109 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 110 | // Method overriding |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 111 | public String toString() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 112 | return super.toString() + " " + salary ; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 113 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 114 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 115 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 116 | public class VirtualDemo { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 117 | public static void main(String [] args) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 118 | Salary s = new Salary("Mohd Mohtashim", "Ambehta, UP", 3, 3600.00); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 119 | Employee emp1 = new Salary("John Adam", "Boston, MA", 1, 2400.00); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 110 | Employee emp2 = new Salary("John Micheal", "Oslo, NR", 2, 3500.00); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 120 | Employee emp3 = new Salary("Ketharine Micheal", "Baali, BA", 3, 4500.00); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 121 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 122 | List<Employee> employees = new ArrayList<Employee>(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 123 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 124 | employees.add(emp1); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 125 | employees.add(emp2); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 126 | employees.add(emp3); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 127 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 128 | System.out.println("Call mailCheck using Salary reference --");  s.mailCheck(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 129 | emp1.toString(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 130 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 131 | System.out.println("\n Call mailCheck using Employee reference--"); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 132 | emp1.mailCheck(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 133 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 134 | // Association between both the  // classes in main method |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 135 | Bank bank = new Bank("Axis Bank",employees); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 136 | System.out.println(emp1.getName() +  " is employee of " + bank.getBankName()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 137 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 138 | // Java program to illustrate  // the concept of Composition |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 139 | List<Employee> emps = bank.getTotalEmployeesInBank(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 140 | for(Employee em : emps){ |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 141 | System.out.println("Name : " + em.getName() + " and "  +" Number : " + em.getNumber()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 142 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 143 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 144 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  | **Total** |  |  |  |  |  |  |  |  |  |  | |  |  |  |

## 

This program considers the Employee as the target class.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Line No** | **Java Program** | **Inheritance** | | | **Polymorphism** | | | **Association** | | | | **Composition** | | | **Total** |
| **If** | **Wl** | **nwl** | **pf** | **wp** | **nwp** | **Af** | **wA** | **nwA** | **Cf** | | **wC** | **nwC** |  |
| 1 | import java.io. \*; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 2 | import java.util.\*; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 3 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 4 | // class bank |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 5 | class Bank { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 6 | private String name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 7 | private final List<Employee> employees; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 8 | static String location = "Colombo"; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 9 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 10 | Bank (String name, List<Employee> employees) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 11 | this.name = name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 12 | this.employees = employees; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 13 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 14 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 15 | public String getBankName() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 16 | return this.name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 17 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 18 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 19 | public List<Employee> getTotalEmployeesInBank(){ |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 20 | return employees; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 21 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 22 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 23 | static class StaticNestedClass { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 24 | void display() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 25 | System.out.println("owner is: " + owner); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 26 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 27 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 28 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 29 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 30 | class Employee { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 31 | private String name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 32 | private String address; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 33 | private int number; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 34 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 35 | public Employee(String name, String address, int number) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 36 | System.out.println("Constructing an Employee"); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 37 | this.name = name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 38 | this.address = address; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 39 | this.number = number; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 40 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 41 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 42 | public void mailCheck() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 43 | System.out.println("Mailing a check to " + this.name + " " + this.address); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 44 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 45 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 46 | public String toString() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 47 | return name + " " + address + " " + number; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 48 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 49 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 50 | public String getName() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 51 | return name; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 52 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 53 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 54 | public String getAddress() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 55 | return address; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 56 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 57 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 58 | public void setAddress(String newAddress) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 59 | address = newAddress; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 60 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 61 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 62 | public int getNumber() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 63 | return number; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 64 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 65 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 66 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 67 | class Salary extends Employee { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 68 | private double salary; // Annual salary |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 69 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 70 | public Salary(String name, String address, int number, double salary) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 71 | super(name, address, number); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 72 | setSalary(salary); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 73 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 74 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 75 | public void mailCheck() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 76 | System.out.println("Within mailCheck of Salary class "); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 77 | System.out.println("Mailing check to " + getName()  + " with salary " + salary); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 78 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 79 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 80 | public double getSalary() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 81 | return salary; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 82 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 83 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 84 | public void setSalary(double newSalary) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 85 | if(newSalary >= 0.0) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 86 | salary = newSalary; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 87 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 88 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 89 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 90 | //Method overloading |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 91 | public void setSalary(double newSalary, int months) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 92 | months = 12; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 93 | System.out.println(" Annual salary pay for: " + getName()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 94 | salary = newSalary \* 12; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 95 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 96 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 97 | public double computePay() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 98 | System.out.println("Computing salary pay for " + getName()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 99 | return salary/52; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 100 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 101 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 102 | //Method overloading |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 103 | public double computePay(double newSalary, double ETF) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 104 | ETF = 1000.00; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 105 | System.out.println("Computing salary pay for " + getNumber()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 106 | newSalary = salary - ETF; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 107 | return newSalary; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 108 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 109 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 110 | // Method overriding |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 111 | public String toString() { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 112 | return super.toString() + " " + salary ; |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 113 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 114 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 115 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 116 | public class VirtualDemo { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 117 | public static void main(String [] args) { |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 118 | Salary s = new Salary("Mohd Mohtashim", "Ambehta, UP", 3, 3600.00); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 119 | Employee emp1 = new Salary("John Adam", "Boston, MA", 1, 2400.00); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 110 | Employee emp2 = new Salary("John Micheal", "Oslo, NR", 2, 3500.00); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 120 | Employee emp3 = new Salary("Ketharine Micheal", "Baali, BA", 3, 4500.00); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 121 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 122 | List<Employee> employees = new ArrayList<Employee>(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 123 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 124 | employees.add(emp1); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 125 | employees.add(emp2); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 126 | employees.add(emp3); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 127 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 128 | System.out.println("Call mailCheck using Salary reference --");  s.mailCheck(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 129 | emp1.toString(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 130 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 131 | System.out.println("\n Call mailCheck using Employee reference--"); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 132 | emp1.mailCheck(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 133 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 134 | // Association between both the  // classes in main method |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 135 | Bank bank = new Bank("Axis Bank",employees); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 136 | System.out.println(emp1.getName() +  " is employee of " + bank.getBankName()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 137 |  |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 138 | // Java program to illustrate  // the concept of Composition |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 139 | List<Employee> emps = bank.getTotalEmployeesInBank(); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 140 | for(Employee em : emps){ |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 141 | System.out.println("Name : " + em.getName() + " and "  +" Number : " + em.getNumber()); |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 142 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 143 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
| 144 | } |  |  |  |  |  |  |  |  |  |  | |  |  |  |
|  | **Total** |  |  |  |  |  |  |  |  |  |  | |  |  |  |

### 6.4 Program Four

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Line No** | **Statement** | **Inheritance** | | | **Polymorphism** | | | **Association** | | | **Composition** | | | **Total** |
| **If** | **Wl** | **nwl** | **pf** | **wp** | **nwp** | **Af** | **wA** | **nwA** | **Cf** | **wC** | **nwC** |
|  | import java.util.ArrayList; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | import java.util.List; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | class Pet { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | private String name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | private int age; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | static String owner = "Shena"; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public Pet(String name, int age) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.name = name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.age = age; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public String getName() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | return name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public void setName(String name) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.name = name; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public int getAge() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | return age; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public void setAge(int age) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.age = age; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | return("Name is: "+getName()  +"\n"  + "Age is:"+getAge()); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | static class StaticNestedClass { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | void display() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | System.out.println("owner is: " + owner); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | class Cat extends Pet { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | private String breed; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public Cat(String name, int age, String breed) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | super(name,age); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.breed = breed; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.setName(name); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.setAge(age); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public String getBreed() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | return breed; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public void setBreed(String breed) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.breed = breed; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public void setPetPrice(double price){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | price = 18000.00; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public void setPetPrice(double price, double specialBreedPrice){ |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | specialBreedPrice = 10000.00; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | price = 18000.00 + specialBreedPrice; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | @Override |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public String toString() { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | return (super.toString()+  "\nBreed is: "+getBreed()); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | //Composition |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | class Owner { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | private final List<Cat> cats; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public Owner(List<Cat> cats) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | this.cats = cats; |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public void addCat(Cat cat) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | cats.add(cat); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public class Main { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | public static void main(String[] args) { |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Cat c1 = new Cat("Kitty",3,"Persian"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Cat c2 = new Cat("Sammy",5,"Bengal"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Cat c3 = new Cat("Misty",2,"Siamese"); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | List<Cat> cats = new ArrayList<>(); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | cats.add(c1); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | cats.add(c2); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | cats.add(c3); |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | } |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| **Total** | | | | | | | | | | | | | |  |

# 7.0 Conclusion

# 8.0 References