Department of Computer Science & Engineering
UNIVERSITY OF LIBERAL ARTS
BANGLADESH

Lab Report 6

Submitted to:
Shakib Mahmud Dipto
Faculty

Submitted by:
Sumaiya Akter
ID: 201014071

Department of CSE
Summer'24

Course code: CSE 2104
Course Title: Object Oriented Programming Lab
Section: 01

University of Liberal Arts Bangladesh

August 25, 2024

Problem 01

Code Explanation :

The Car class represents a car with properties such as make, model, year, color, and price. It provides getter and setter methods for each property to encapsulate the data. The main method creates three car objects, sets their properties using the setter methods, and prints their details using the getter methods. The corrected code fixes issues with the getColor method signature and return statement, and syntax errors in the print statements.

Code Screenshots:

Input:

Car.java × Person.java × Employee.java ×

Source History

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to cha
 */

package car;

/**
 *
 * @author ANIK
 */
public class Car {
    private String make;
    private String model;
    private int year;
    private String color;
    private double price;

    public String getMake() {
        return make;
    }

    public void setMake(String make) {
        this.make = make;
    }

    public String getModel() {
        return model;
    }

    public void setModel(String model) {
        this.model = model;
    }

    public int getYear() {
        return year;
    }

    public void setYear(int year) {
        this.year = year;
    }

    public String getColor() { // Removed parameter
        return color; // Added return statement
    }

    public void setColor(String color) { // Added setColor method
        this.color = color;
    }
```

```
49
50      public double getPrice() {
51          return price;
52      }
53
54      public void setPrice(double price) { // Added setPrice method
55          this.price = price;
56      }
57
58      public static void main(String[] args) {
59          Car car1 = new Car();
60          car1.setMake("Toyota");
61          car1.setModel("Camry");
62          car1.setYear(2022);
63          car1.setColor("Silver");
64          car1.setPrice(25000.00);
65
66          Car car2 = new Car();
67          car2.setMake("Honda");
68          car2.setModel("Accord");
69          car2.setYear(2021);
70          car2.setColor("Red");
71          car2.setPrice(28000.00);
72
73          Car car3 = new Car();
74          car3.setMake("Ford");
75          car3.setModel("Mustang");
76          car3.setYear(2023);
77          car3.setColor("Blue");
78          car3.setPrice(35000.00);
79
80          System.out.println("Car 1 :");
81          System.out.println("Make : " + car1.getMake());
82          System.out.println("Model : " + car1.getModel());
83          System.out.println("Year : " + car1.getYear());
84          System.out.println("Color: " + car1.getColor()); // Fixed concatenation
85          System.out.println("Price: $ " + car1.getPrice()); // Fixed method call
86          System.out.println();
87
88          System.out.println("Car 2 : ");
89          System.out.println("Make : " + car2.getMake());
90          System.out.println("Model : " + car2.getModel());
91          System.out.println("Year : " + car2.getYear());
92          System.out.println("Color: " + car2.getColor()); // Fixed concatenation
93          System.out.println("Price: $ " + car2.getPrice()); // Fixed method call
94          System.out.println();
```

```
96              System.out.println("Car 3 : ");
97              System.out.println("Make : " + car3.getMake());
98              System.out.println("Model : " + car3.getModel());
99              System.out.println("Year : " + car3.getYear());
100             System.out.println("Color: " + car3.getColor()); // Fixed concatenation
101             System.out.println("Price: $ " + car3.getPrice()); // Fixed method call
102             System.out.println();
103         }
104     }
```

Output:

```
Output - Run (Car)
cd F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 6\codes\Car; "JAVA_HOME=C:\\Program File
Scanning for projects...

------------------------< com.mycompany:Car >-------------------------
Building Car 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Car ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 6\codes

--- compiler:3.11.0:compile (default-compile) @ Car ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ Car ---
Car 1 :
Make : Toyota
Model : Camry
Year : 2022
Color: Silver
Price: $ 25000.0

Car 2 :
Make : Honda
Model : Accord
Year : 2021
Color: Red
Price: $ 28000.0

Car 3 :
Make : Ford
Model : Mustang
Year : 2023
Color: Blue
Price: $ 35000.0

------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.554 s
Finished at: 2024-08-25T13:13:55+06:00
------------------------------------------------------------------------
```

Practice Problem 01

Code Explanation:

The Person class encapsulates the properties of a person, including name, age, gender, and address. The class provides getter and setter methods for each property to ensure data integrity and accessibility. The main method creates a Person object, sets its properties using the setter methods, and prints the person's details using the getter methods. This code demonstrates the use of classes, objects, and encapsulation in Java.

Code Screenshot:

Input:

Car.java | Person.java | Employee.java

Source | History

```
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this l
 */

package person;

/**
 *
 * @author ANIK
 */
public class Person  {
    private String name;
    private int age;
    private String gender;
    private String address;

    // Getter and Setter for name
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    // Getter and Setter for age
    public int getAge() {
        return age;
    }

    public void setAge(int age) {
        this.age = age;
    }

    // Getter and Setter for gender
    public String getGender() {
        return gender;
    }

    public void setGender(String gender) {
        this.gender = gender;
    }

    // Getter and Setter for address
    public String getAddress() {
        return address;
    }
}
```

```
48
49      public void setAddress(String address) {
50          this.address = address;
51      }
52
53      public static void main(String[] args) {
54          // Example usage
55          Person person = new Person();
56          person.setName("John Doe");
57          person.setAge(30);
58          person.setGender("Male");
59          person.setAddress("123 Main St, Anytown, USA");
60
61          // Displaying person details
62          System.out.println("Name: " + person.getName());
63          System.out.println("Age: " + person.getAge());
64          System.out.println("Gender: " + person.getGender());
65          System.out.println("Address: " + person.getAddress());
66      }
67  }
```

Output:

```
Output - Run (Person)
   cd F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 6\codes\Person; "JAVA_HOME=C:\\Progr
   Scanning for projects...

   ------------------------< com.mycompany:Person >------------------------
   Building Person 1.0-SNAPSHOT
     from pom.xml
   --------------------------------[ jar ]---------------------------------

   --- resources:3.3.1:resources (default-resources) @ Person ---
   skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 6\c

   --- compiler:3.11.0:compile (default-compile) @ Person ---
   Nothing to compile - all classes are up to date

   --- exec:3.1.0:exec (default-cli) @ Person ---
   Name: John Doe
   Age: 30
   Gender: Male
   Address: 123 Main St, Anytown, USA
   ------------------------------------------------------------------------
   BUILD SUCCESS
   ------------------------------------------------------------------------
   Total time:  0.541 s
   Finished at: 2024-08-25T13:15:13+06:00
   ------------------------------------------------------------------------
```

Practice Problem 02

Code Explanation:

The Employee class models an employee with properties such as name, ID, salary, and designation. It includes getter and setter methods for each property, allowing controlled access and modification of the data. The main method demonstrates the creation of an Employee object, setting its properties using the setter methods, and displaying the employee's details through the getter methods. This implementation showcases encapsulation and object-oriented programming principles in Java.

Code Screenshot:

Input:

Car.java × Person.java × Employee.java ×

Source History

```java
/*
 * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to
 */

package employee;

/**
 *
 * @author ANIK
 */
public class Employee {
    private String name;
    private int id;
    private double salary;
    private String designation;

    // Getter and Setter for name
    public String getName() {
        return name;
    }

    public void setName(String name) {
        this.name = name;
    }

    // Getter and Setter for id
    public int getId() {
        return id;
    }

    public void setId(int id) {
        this.id = id;
    }

    // Getter and Setter for salary
    public double getSalary() {
        return salary;
    }

    public void setSalary(double salary) {
        this.salary = salary;
    }

    // Getter and Setter for designation
    public String getDesignation() {
        return designation;
    }
```

```
48
49      // Method to update designation
50      public void updateDesignation(String designation) {
51          this.designation = designation;
52      }
53
54      // Method to update both salary and designation
55      public void updateSalaryAndDesignation(double salary, String designation) {
56          this.salary = salary;
57          this.designation = designation;
58      }
59
60      // Main method for testing
61      public static void main(String[] args) {
62          Employee emp1 = new Employee("Alice", 101, 75000, "Software Engineer");
63          Employee emp2 = new Employee("Bob", 102);
64          Employee emp3 = new Employee("Charlie");
65
66          emp1.displayEmployeeInfo();
67          emp2.displayEmployeeInfo();
68          emp3.displayEmployeeInfo();
69
70          // Updating salary and designation for emp1
71          emp1.updateSalary(80000);
72          emp1.updateDesignation("Senior Software Engineer");
73
74          System.out.println("\nUpdated Information for Employee 1:");
75          emp1.displayEmployeeInfo();
76      }
77  }
```

Output:

Output - Run (Employee)

```
cd F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab 5\code\Employee; "JAVA_HOME=C:\\P
Scanning for projects...

-----------------------< com.mycompany:Employee >-----------------------
Building Employee 1.0-SNAPSHOT
  from pom.xml
--------------------------------[ jar ]---------------------------------

--- resources:3.3.1:resources (default-resources) @ Employee ---
skip non existing resourceDirectory F:\sumaiya the V.I.P\sem 14\OOP\assignments\lab

--- compiler:3.11.0:compile (default-compile) @ Employee ---
Nothing to compile - all classes are up to date

--- exec:3.1.0:exec (default-cli) @ Employee ---
Employee Information:
Name: Alice
ID: 101
Salary: $75000.0
Designation: Software Engineer
Employee Information:
Name: Bob
ID: 102
Salary: $0.0
Designation: Unknown
Employee Information:
Name: Charlie
ID: 0
Salary: $0.0
Designation: Unknown

Updated Information for Employee 1:
Employee Information:
Name: Alice
ID: 101
Salary: $80000.0
Designation: Senior Software Engineer
------------------------------------------------------------------------
BUILD SUCCESS
------------------------------------------------------------------------
Total time:  0.672 s
Finished at: 2024-08-25T13:16:31+06:00
------------------------------------------------------------------------
```