

# Assignment 2

**Aim:** Implementation of a given problem statement/s using classes and objects

Que.1) Create a class called Employee that includes three pieces of information as instance variables: first name, last name, and monthly salary. Your class should have a constructor that initializes the three instance variables. Provide a setter and getter method for each instance variable. If the monthly salary is not positive, set it to 0.0. Write a test application named EmployeeTest that demonstrates the Employee class's capabilities. Create two Employee objects and display each object's yearly salary. Then give each Employee a 10% raise and display each Employee's yearly salary again

→

```
class Employee {  
    private String firstName;  
    private String lastName;  
    private double monthlySalary;  
  
    public Employee(String firstName, String lastName, double monthlySalary) {  
        this.firstName = firstName;  
        this.lastName = lastName;  
        this.monthlySalary = Math.max(monthlySalary, 0.0);  
    }  
  
    public String getFirstName() {  
        return firstName;  
    }  
  
    public void setFirstName(String firstName) {  
        this.firstName = firstName;  
    }  
  
    public String getLastName() {
```

```
    return lastName;
}
```

```
public void setLastName(String lastName) {
    this.lastName = lastName;
}
```

```
public double getMonthlySalary() {
    return monthlySalary;
}
```

```
public void setMonthlySalary(double monthlySalary) {
    this.monthlySalary = Math.max(monthlySalary, 0.0);
}
```

```
public double getYearlySalary() {
    return monthlySalary * 12;
}
```

```
public void giveRaise(double percentage) {
    this.monthlySalary += this.monthlySalary * (percentage / 100);
}
}
```

```
public class EmployeeTest {
    public static void main(String[] args) {
        Employee emp1 = new Employee("John", "Doe", 3000);
        Employee emp2 = new Employee("Jane", "Smith", 4000);
    }
}
```

```
        System.out.printf("%s %s Yearly Salary: $%.2f%n", emp1.getFirstName(),  
emp1.getLastName(), emp1.getYearlySalary());
```

```
        System.out.printf("%s %s Yearly Salary: $%.2f%n", emp2.getFirstName(),  
emp2.getLastName(), emp2.getYearlySalary());
```

```
        emp1.giveRaise(10);
```

```
        emp2.giveRaise(10);
```

```
        System.out.println("After 10% raise:");
```

```
        System.out.printf("%s %s Yearly Salary: $%.2f%n", emp1.getFirstName(),  
emp1.getLastName(), emp1.getYearlySalary());
```

```
        System.out.printf("%s %s Yearly Salary: $%.2f%n", emp2.getFirstName(),  
emp2.getLastName(), emp2.getYearlySalary());
```

```
    }
```

```
}
```

---

Que.2) Implement a Java program to print the area of a rectangle by creating a class named 'Area' that has two methods. The first method, named 'setDim', takes the length and breadth of the rectangle as parameters. The second method, named 'getArea', returns the area of the rectangle. The length and breadth of the rectangle are entered through the keyboard.

→

```
import java.util.Scanner;
```

```
class Area {
```

```
    private double length;
```

```
    private double breadth;
```

```
    public void setDim(double length, double breadth) {
```

```
        this.length = length;
```

```

        this.breadth = breadth;
    }

    public double getArea() {
        return length * breadth;
    }
}

public class AreaTest {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        Area rectangle = new Area();

        System.out.print("Enter length: ");
        double length = scanner.nextDouble();

        System.out.print("Enter breadth: ");
        double breadth = scanner.nextDouble();

        rectangle.setDim(length, breadth);
        System.out.println("Area of the rectangle: " + rectangle.getArea());

        scanner.close();
    }
}

```

---

Que.3) Write a Java program to demonstrate the use of static variables, static blocks, and static methods

→

```

class StaticDemo {
    static int staticVariable;

    static {
        staticVariable = 100;
        System.out.println("Static block executed. Static variable initialized to " + staticVariable);
    }

    static void staticMethod() {
        System.out.println("Static method called. Static variable value: " + staticVariable);
    }
}

public class StaticTest {
    public static void main(String[] args) {
        System.out.println("Main method starts");
        StaticDemo.staticMethod();
        System.out.println("Changing static variable value to 200");
        StaticDemo.staticVariable = 200;
        StaticDemo.staticMethod();
    }
}

```

---

Que.4) Write a Java program to implement a stack and a queue

→

```

import java.util.LinkedList;
import java.util.Queue;

```

```
import java.util.Stack;

class StackExample {
    private Stack<Integer> stack = new Stack<>();

    public void push(int value) {
        stack.push(value);
        System.out.println(value + " pushed to stack");
    }

    public int pop() {
        if (stack.isEmpty()) {
            System.out.println("Stack is empty");
            return -1;
        }
        return stack.pop();
    }
}

class QueueExample {
    private Queue<Integer> queue = new LinkedList<>();

    public void enqueue(int value) {
        queue.add(value);
        System.out.println(value + " added to queue");
    }

    public int dequeue() {
        if (queue.isEmpty()) {
```

```

        System.out.println("Queue is empty");
        return -1;
    }
    return queue.poll();
}
}

public class StackQueueTest {
    public static void main(String[] args) {
        StackExample stack = new StackExample();
        stack.push(10);
        stack.push(20);
        System.out.println("Popped from stack: " + stack.pop());

        QueueExample queue = new QueueExample();
        queue.enqueue(30);
        queue.enqueue(40);
        System.out.println("Dequeued from queue: " + queue.dequeue());
    }
}

```

---

Que.5) Write a Java program to arrange 10 names in alphabetical order.

→

```

import java.util.Arrays;
import java.util.Scanner;

public class NameSorter {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
    }
}

```

```
String[] names = new String[10];

System.out.println("Enter 10 names:");
for (int i = 0; i < 10; i++) {
    names[i] = scanner.nextLine();
}

Arrays.sort(names);

System.out.println("Names in alphabetical order:");
for (String name : names) {
    System.out.println(name);
}

scanner.close();
}
```

---

## **Conclusion:**

Students should be able to write Java programs using classes and objects after completing this content