Lab 1-9

LAB_1

Q.1

Write a Java program to display “Hello World”.

Ans.

```
package inheritance;

public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("Hello World");
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Hello World
Process finished with exit code 0

Q.2

Write a Java program to print numbers between 1 to n which are divisible by 3, 5 and by both(3 and 5) by taking n as an input from the user

Ans.

```
package inheritance;

import java.util.Scanner;

public class DivisibleByThreeAndFive {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the value of n: ");
      int n = scanner.nextInt();
      scanner.close();

      System.out.println("Numbers between 1 and " + n + " divisible by 3, 5, and both:");

      for (int i = 1; i <= n; i++) {
         if (i % 3 == 0 && i % 5 == 0) {
            System.out.println(i + " (divisible by 3 and 5)");
         } else if (i % 3 == 0) {
            System.out.println(i + " (divisible by 3)");
         } else if (i % 5 == 0) {
            System.out.println(i + " (divisible by 5)");
         }
      }
   }
}
```

Output:

Enter the value of n: 10
Numbers between 1 and 10 divisible by 3, 5, and both:
3 (divisible by 3)
5 (divisible by 5)
6 (divisible by 3)
9 (divisible by 3)

10 (divisible by 5)
Process finished with exit code 0

Q.3

Write a class named Greeter that prompts the user for his or her name, and then prints a personalized greeting. As an example, if the user entered "Era", the program should respond "Hello Era!".

Ans.

```
import java.util.Scanner;

public class Greeter {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter your name: ");
      String userName = scanner.nextLine();
      scanner.close();
      System.out.println("Hello " + userName + "!");
   }
}
```

Output:

Enter your name: Hello lakhman!

Q.4

Write a Java program that takes Name, Roll No and marks of 5 subjects as input and gives a formatted output as:

Name: ABCD

Roll No. : 1

Average: 84

Also display the grade (e.g. A, B, C. . . etc) using the average.

Ans.

```
package inheritance;

import java.util.Scanner;

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

      System.out.print("Enter Name: ");
      String name = scanner.nextLine();

      System.out.print("Enter Roll No.: ");
      int rollNo = scanner.nextInt();

      System.out.print("Enter marks for 5 subjects (separated by spaces): ");
      int marks1 = scanner.nextInt();
      int marks2 = scanner.nextInt();
      int marks3 = scanner.nextInt();
      int marks4 = scanner.nextInt();
      int marks5 = scanner.nextInt();

      scanner.close();

      int totalMarks = marks1 + marks2 + marks3 + marks4 + marks5;
      double average = totalMarks / 5.0;

      System.out.println("Name: " + name);
      System.out.println("Roll No.: " + rollNo);
      System.out.println("Average: " + (int) average);

      if (average >= 90) {
         System.out.println("Grade: A");
      } else if (average >= 80) {
         System.out.println("Grade: B");
```

```
        } else if (average >= 70) {
            System.out.println("Grade: C");
        } else if (average >= 60) {
            System.out.println("Grade: D");
        } else {
            System.out.println("Grade: F");
        }
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter Name: lakhman
Enter Roll No.: 30
Enter marks for 5 subjects (separated by spaces): 80 90 96 89 97
Name: Lakhman
Roll No.: 30
Average: 90
Grade: A
Process finished with exit code 0

Q.5

Calculate and return the sum of all the even numbers present in the numbers array passed to the method calculateSumOfEvenNumbers. Implement the logic inside calculateSumOfEvenNumbers() method.

Test the functionalities using the main() method of the Tester class.

Test the functionalities using the main() method of the Tester class.

Sample Input and Output:

Sample Input	Sample Output
{68,79,86,99,23,2,41,100}	256
{1,2,3,4,5,6,7,8,9,10}	30

Ans.

```
package inheritance;

public class Tester {
    public static void main(String[] args) {
        int[] numbers1 = {68, 79, 86, 99, 23, 2, 41, 100};
        int[] numbers2 = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10};

        int sum1 = calculateSumOfEvenNumbers(numbers1);
        int sum2 = calculateSumOfEvenNumbers(numbers2);

        System.out.println(sum1);
        System.out.println(sum2);
    }

    public static int calculateSumOfEvenNumbers(int[] numbers) {
        int sum = 0;

        for (int number : numbers) {
            if (number % 2 == 0) {
                sum += number;
            }
        }

        return sum;
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
256
30
Process finished with exit code 0

Q.6

1. Write a program to perform matrix addition and matrix multiplication on two given matrices. Use for each form of for loop to display the matrices.

Ans.

```
public class MatrixOperations {
  public static void main(String[] args) {
    int[][] matrix1 = {
      {1, 2, 3},
      {4, 5, 6},
      {7, 8, 9}
    };

    int[][] matrix2 = {
      {9, 8, 7},
      {6, 5, 4},
      {3, 2, 1}
    };

    System.out.println("Matrix 1:");
    displayMatrix(matrix1);

    System.out.println("nMatrix 2:");
    displayMatrix(matrix2);

    System.out.println("nMatrix Addition:");
    int[][] sumMatrix = addMatrices(matrix1, matrix2);
    displayMatrix(sumMatrix);

    System.out.println("nMatrix Multiplication:");
    int[][] productMatrix = multiplyMatrices(matrix1, matrix2);
    displayMatrix(productMatrix);
  }

  public static void displayMatrix(int[][] matrix) {
    for (int[] row : matrix) {
      for (int element : row) {
```

```
            System.out.print(element + " ");
        }
        System.out.println();
    }
}

public static int[][] addMatrices(int[][] matrix1, int[][] matrix2) {
    int rows = matrix1.length;
    int columns = matrix1[0].length;
    int[][] resultMatrix = new int[rows][columns];

    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < columns; j++) {
            resultMatrix[i][j] = matrix1[i][j] + matrix2[i][j];
        }
    }

    return resultMatrix;
}

public static int[][] multiplyMatrices(int[][] matrix1, int[][] matrix2) {
    int rows1 = matrix1.length;
    int columns1 = matrix1[0].length;
    int columns2 = matrix2[0].length;
    int[][] resultMatrix = new int[rows1][columns2];

    for (int i = 0; i < rows1; i++) {
        for (int j = 0; j < columns2; j++) {
            for (int k = 0; k < columns1; k++) {
                resultMatrix[i][j] += matrix1[i][k] * matrix2[k][j];
            }
        }
    }

    return resultMatrix;
}
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Matrix 1:
1 2 3
4 5 6
7 8 9
Matrix 2:
9 8 7
6 5 4
3 2 1
Matrix Addition:
10 10 10
10 10 10
10 10 10
Matrix Multiplication:
30 24 18
84 69 54
138 114 90
Process finished with exit code 0

Q.7

Practice Problems:

- Given an integer numRows, return the first numRows of Pascal's triangle using a jagged 2D array In Pascal's triangle, each number is the sum of the two numbers directly above it.

Sample Input and Output:

Sample Input	Sample Output
numRows = 5	[[1],[1,1],[1,2,1],[1,3,3,1],[1,4,6,4,1]]
numRows = 1	[[1]]

Ans.

```
public class PascalsTriangle {
    public static void main(String[] args) {
        int numRows = 5;
        int[][] pascalsTriangle = generatePascalsTriangle(numRows);

        for (int[] row : pascalsTriangle) {
            for (int num : row) {
                System.out.print(num + " ");
            }
            System.out.println();
        }
    }

    public static int[][] generatePascalsTriangle(int numRows) {
        if (numRows <= 0) {
            return new int[0][0];
        }

        int[][] triangle = new int[numRows][];
        for (int i = 0; i < numRows; i++) {
            triangle[i] = new int[i + 1];
            triangle[i][0] = 1;

            for (int j = 1; j < i; j++) {
                triangle[i][j] = triangle[i - 1][j - 1] + triangle[i - 1][j];
            }

            triangle[i][i] = 1;
        }

        return triangle;
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
1
1 1
121
1 3 31
1 4 6 41
Process finished with exit code 0

Q.8

- Write a Java program to convert the integer entered by the user into a roman numeral. Roman numerals are represented by seven different symbols: I, V, X, L, C, D and M.

Symbol	Value
I	1
V	5
X	10
L	50
C	100
D	500
M	1000

Ans.

```
import java.util.Scanner;

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

        System.out.print("Enter an integer: ");
        int num = scanner.nextInt();

        scanner.close();

        if (num < 1 || num > 3999) {
```

```
            System.out.println("Please enter an integer between 1 and 3999.");
        } else {
            String romanNumeral = intToRoman(num);
            System.out.println("Roman numeral: " + romanNumeral);
        }
    }

    public static String intToRoman(int num) {
        int[] values = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};
        String[] symbols = {"M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"};

        StringBuilder result = new StringBuilder();

        for (int i = 0; i < values.length; i++) {
            while (num >= values[i]) {
                num -= values[i];
                result.append(symbols[i]);
            }
        }

        return result.toString();
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter an integer: 200
Roman numeral: CC
Process finished with exit code 0

Q.9

- A group of MIT friends decide to run the Boston Marathon. Their names and times (in minutes) are below: Name Time (minutes) Elena 341 Thomas 273 Hamilton 278 Suzie 329 Phil 445 Matt 402 Alex 388 Emma 275 John 243 James 334 Jane 412 Emily 393 Daniel 299 Neda 343 Aaron 317 Kate 265 Find the fastest runner. Print the name and his/her time (in minutes).

Optional: Find the second fastest runner. Print the name and his/her time (in minutes).

Ans.

```
package inheritance;

public class BostonMarathon {
  public static void main(String[] args) {
    String[] names = {"Elena", "Thomas", "Hamilton", "Suzie", "Phil", "Matt", "Alex", "Emma",
"John", "James", "Jane", "Emily", "Daniel", "Neda", "Aaron", "Kate"};
    int[] times = {341, 273, 278, 329, 445, 402, 388, 275, 243, 334, 412, 393, 299, 343, 317, 265};

    String fastestRunner = findFastestRunner(names, times);
    System.out.println(fastestRunner);

    String secondFastestRunner = findSecondFastestRunner(names, times);
    System.out.println(secondFastestRunner);
  }

  public static String findFastestRunner(String[] names, int[] times) {
    String fastestRunner = "";
    int fastestTime = Integer.MAX_VALUE;

    for (int i = 0; i < times.length; i++) {
      if (times[i] < fastestTime) {
        fastestTime = times[i];
        fastestRunner = names[i];
      }
    }

    return "Fastest Runner: " + fastestRunner + ", " + fastestTime + " minutes";
  }

  public static String findSecondFastestRunner(String[] names, int[] times) {
    String fastestRunner = "";
```

```
        int fastestTime = Integer.MAX_VALUE;
        String secondFastestRunner = "";
        int secondFastestTime = Integer.MAX_VALUE;

        for (int i = 0; i < times.length; i++) {
            if (times[i] < fastestTime) {
                secondFastestTime = fastestTime;
                secondFastestRunner = fastestRunner;

                fastestTime = times[i];
                fastestRunner = names[i];
            } else if (times[i] < secondFastestTime) {
                secondFastestTime = times[i];
                secondFastestRunner = names[i];
            }
        }

        return "Second Fastest Runner: " + secondFastestRunner + ", Time: " + secondFastestTime + "
minutes";
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Fastest Runner: John, Time: 243 minutes
Second Fastest Runner: Kate, Time: 265 minutes
Process finished with exit code 0

LAB_2

Q.1

1. Write a program that returns the number of times that the string “hi” appears anywhere in the given string.

Ans.

```
public class HiCount {

  public static int countHiOccurrences(String inputString) {
    String lowerCaseInput = inputString.toLowerCase();
    int count = 0;

    int index = lowerCaseInput.indexOf("hi");
    while (index != -1) {
      count++;
      index = lowerCaseInput.indexOf("hi", index + 2);
    }

    return count;
  }

  public static void main(String[] args) {
    String inputString = "hi there, hi how are you? Hi, this is a test.";
    int result = countHiOccurrences(inputString);
    System.out.println("The substring"hi" appears " + result + " times in the given string.");
  }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=57141:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -

Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/javabasics/out/production/java

basics lab2.HiCount
The substring "hi" appears 4 times in the given string.

Process finished with exit code 0

Q.2

1. Write a program which checks whether the input string is palindrome or not and then display an appropriate message [e.g. “Refer” is a palindrome string].

Ans.

```
import java.util.Scanner;

public class PalindromeChecker {

    public static boolean isPalindrome(String inputString) {
        String cleanedInput = inputString.replaceAll("s", "").toLowerCase();
        String reversedString = "";

        for (int i = cleanedInput.length() - 1; i >= 0; i–) {
            reversedString += cleanedInput.charAt(i);
        }

        return cleanedInput.equals(reversedString);
    }

    public static void main(String[]ga) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String inputString = scanner.nextLine();

        if (isPalindrome(inputString)) {
            System.out.println("""" + inputString + "" is a palindrome string.");
```

```
        } else {
            System.out.println("""" + inputString + "" is not a palindrome string.");
        }

        scannerclose();
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter a string: yey
"yey" is a palindrome.
Process finished with exit code 0

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter a string: lakhman
"Zakhman" is not a palindrome.
Process finished with exit code 0

Q.3

1. Write a program that takes your full name as input and displays the abbreviations of the first and middle names except the last name which is displayed as it is. For example, if your name is Robert Brett Roser, then the output should be R.B.Roser

Ans.

```
import java.util.Scanner;

public class NameAbbreviation {

    public static String abbreviateName(String fullName) {
        String[] names = fullName.split("s");
        StringBuilder abbreviation = new StringBuilder();

        for (int i = 0; i < names.length - 1; i++) {
            abbreviation.append(names[i].charAt(0)).append(".");
        }

        abbreviation.append(names[names.length - 1]);

        return abbreviation.toString();
    }

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

        System.out.print("Enter your full name: ");
        String fullName = scanner.nextLine();

        System.out.println("Abbreviation: " + abbreviateName(fullName));

        scanner.close();
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter your full name: parmar lakhman jivabhai
Abbreviation: p.l. jivabhai
Process finished with exit code 0

Q.4

1. Write a method **String removeWhiteSpaces(String str)** method that removes all the white spaces from the string passed to the method and returns the modified String. Test the functionalities using the main() method of the Tester class.

Ans.

```
public class Tester {

  public static String removeWhiteSpaces(String str) {
    return str.replaceAll("s", "");
  }

  public static void main(String[] args) {
    String inputString = "This is a test string with white spaces.";

    System.out.println("Original String: " + inputString);
    System.out.println("Modified String: " + removeWhiteSpaces(inputString));
  }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Original String: This is a test string with white spaces.
Modified String: Thisisateststringwithwhitespaces.
Process finished with exit code 0

Q.5

Write a class Student with member variables int roll_no, String name and an array to store marks of 5 subjects. Demonstrate constructor overloading and use this keyword. Write a findAverage() method that returns double value. Write a TestStudent class containing main() method to do the following:

Store the details of one student by creating one object of Student class and display them.

Store the details of 3 students by creating an array of objects of Student class and display the details of the student who has the highest average amongst the three students.

Ans.

```
public class Student {
   private int rollNo;
   private String name;
   private int[] marks;

   public Student() {
      this.rollNo = 0;
      this.name = "";
      this.marks = new int[5];
   }

   public Student(int rollNo, String name) {
      this();
      this.rollNo = rollNo;
      this.name = name;
   }

   public Student(int rollNo, String name, int[] marks) {
      this(rollNo, name);
      this.marks = marks;
   }

   public double findAverage() {
      double sum = 0;
      for (int mark : marks) {
         sum += mark;
      }
      return sum / marks.length;
   }
```

```
}

public class TestStudent {
    public static void main(String[] args) {
        Student student1 = new Student();
        System.out.println("Student 1 average: " + student1.findAverage());

        Student student2 = new Student(101, "John Doe");
        System.out.println("Student 2 average: " + student2.findAverage());

        int[] marks = {90, 85, 78, 92, 88};
        Student student3 = new Student(102, "Jane Doe", marks);
        System.out.println("Student 3 average: " + student3.findAverage());
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
1
Lakhman
80 90 90 90 90
2
kunal
90 90 90 90 90
3
dev
89 90 89 78 99
Student with the highest average marks: kunal
Process finished with exit code 0

Practice Problems:

Q.6

- Write a Java program to find the second most frequent character in a given string.

E.g.

Input	Success
Output	c

Ans.

```
import java.util.*;

class Main {
  static void countFreq(String s, int[] freq) {
    for (int i = 0; i < s.length(); i++) {
      int temp = s.charAt(i);
      if (temp >= 97 && temp <= 122) {
        temp = temp - 32;
      }
      freq[temp - 65] = freq[temp - 65] + 1;
    }
  }

  public static void main(String[] args) {
    Scanner scanner = new Scanner(System.in);
    int[] freq = new int[26];
    for (int i = 0; i < 26; i++) {
      freq[i] = 0;
    }
    String item = scanner.nextLine();
    countFreq(item, freq);
    int max = 0;
    int index = 0;
    for (int i = 0; i < 26; i++) {
      if (freq[i] > max) {
        max = freq[i];
        index = i;
```

```
        }
      }
      index += 65;
      char x = (char) index;
      System.out.print(x);
      System.out.print(" ");
    }
}
```

Output:

```
Finished in 141 ms
Enter a string:ccccccadfasdfcccc
Most frequent letter: c
Finished in 166 ms
Enter a string:AccccAAAAAAAWFEF
Most frequent letter A
```

Q.7

- Write a program that prints the words of a sentence given as input in reverse order

E.g.

Input	This is a java programming lab
Output	lab programming java a is This

Ans.

```
import java.util.Scanner;

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

      System.out.println("Enter a sentence: ");
```

```
        String sentence = scanner.nextLine();

        String[] words = sentence.split("\s+"); // Split the sentence into words

        System.out.println("Reversed sentence:");
        for (int i = words.length - 1; i >= 0; i--) {
            System.out.print(words[i] + " ");
        }
    }
}
```

Output:

```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -
javaagent:/Users/lakhman/Applications/IntelliJ IDEA
Enter a sentence: my name is lakhman
Reversed sentence: lakhman is name my
Process finished with exit code 0
```

Q.8

- Write a Java program to find the longest palindrome substring within a string.

E.g.

Input	thequickbrownfoxxofnworbquickthe
Output	The longest palindrome substring in the given string is: brownfoxxofnworb The length of the palindromic substring is: 16

Ans.

```
public class LongestPalindromeSubstring {
public static void main(String[] args) {
String input = "thequickbrownfoxxofnworbquickthe";
```

```
String longestPalindrome = findLongestPalindrome(input);

System.out.println("The longest palindrome substring in the given
string is:");
System.out.println(longestPalindrome);
System.out.println("The length of the palindromic substring is: " +

longestPalindrome.length());
}

private static String findLongestPalindrome(String input) {
int maxLength = 0;
String longestPalindrome = "";

for (int i = 0; i < input.length(); i++) {
for (int j = i + 1; j <= input.length(); j++) {
String sub = input.substring(i, j);
if (isPalindrome(sub) && sub.length() > maxLength) {
maxLength = sub.length();
longestPalindrome = sub;
}
}
}

return longestPalindrome;
}

private static boolean isPalindrome(String str) {
int left = 0;
int right = str.length() - 1;

while (left < right) {
if (str.charAt(left) != str.charAt(right)) {
return false;
}
left++;
right--;
```

```
}
return true;
}

}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=57152:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/javabasics/out/production/java basics lab2.LongestPalindromeSubstring

The longest palindrome substring in the given string is:
brownfoxxofnworb
The length of the palindromic substring is: 16

Process finished with exit code

LAB_3

Q.1

Write a Java program that checks for prime number using the object oriented approach. [Hint: create a class NumberClass with a member value and method isPrimeNumber()]

Ans.

```
import java.util.*;
class NumberClass {
private int value;
public NumberClass(int value) {
this.value = value;
}
public boolean isPrimeNumber() {
if (value <= 1) {
return false;
} else {
for (int i = 2; i <= Math.sqrt(value); i++) {
if (value % i == 0) {
return false;
}
}
return true;
}
}
}
class Main {
public static void main(String[]gar) {
Scanner scan=new Scanner(System.in);
int num1=scan.nextInt();
NumberClass numberObj = new NumberClass(num1);
if (numberObj.isPrimeNumber()) {
System.out.println(num1 + " is a prime number");
} else {
System.out.println(num1 + " is not a prime number");
}
}
}
```

Output:

```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -
javaagent:/Users/lakhman/Applications/IntelliJ IDEA
23
```

23 is a prime number
Process finished with exit code 0

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA
22
22 is not a prime number
Process finished with exit code 0

Q.2

1. Create two classes:

class Person

Derive a class Student from class Person.

Person

- name : String
- age : int

+ Person()

+ Person(name : String, age : int)

+ getName() : String

+ getAge() : int

+ setName(name : String) : void

+ setAge(age : int) : void

+ toString() : String

Student

- rollno : int
- marks : double[]

+ Student()

+ Student(rollno : int)

+ Student(rollno : int, marks : double[])

+ Student(rollno : int, name : String, age : int, marks : double[])

+ getRollno() : int

+ getMarks() : double[]

+ setRollno(rollno: int) : void

+ setMarks(marks : double[]) : void

+ toString() : String

+ displayDetails() : void

Add the following to Student class:

- a static variable **count**(to count the number of objects)
- a static block to initialize count variable to zero
- a static method String **getCount()** that returns the number of student objects created
- Write a TestStudent class containing the main() method.
- Store the details of 3 students by creating an array of objects of Student class and display the student who has highest average amongst the three students as follows using **displayDetails()** method for that object:

e.g.

RollNo = 100

Name =ABC

Age = 20

Marks=78 86 88 67 92

- Create one more object of the Student class and then call the getCount() to display the number of Student objects created.

Ans.

```
import java.util.Scanner;

class Person {
   private String name;
   private int age;
```

```
    public Person() {
        this.name = "";
        this.age = 0;
    }

    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

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

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

    public String toString() {
        return "Name: " + name + ", Age: " + age;
    }
}

class Student extends Person {
    private int rollno;
    private double[] marks;
    private static int count = 0;

    static {
```

```
    count = 0;
}

public Student() {
    super();
    this.rollno = 0;
    this.marks = new double[0];
    count++;
}

public Student(int rollno) {
    super();
    this.rollno = rollno;
    this.marks = new double[0];
    count++;
}

public Student(int rollno, double[] marks) {
    super();
    this.rollno = rollno;
    this.marks = marks;
    count++;
}

public Student(int rollno, String name, int age, double[] marks) {
    super(name, age);
    this.rollno = rollno;
    this.marks = marks;
    count++;
}

public int getRollno() {
    return rollno;
}

public double[] getMarks() {
    return marks;
}
```

```
    public void setRollno(int rollno) {
        this.rollno = rollno;
    }

    public void setMarks(double[] marks) {
        this.marks = marks;
    }

    public static String getCount() {
        return "Number of Student objects created: " + count;
    }

    public void displayDetails() {
        System.out.println("RollNo = " + rollno);
        System.out.println("Name = " + getName());
        System.out.println("Age = " + getAge());
        System.out.print("Marks = ");
        for (double mark : marks) {
            System.out.print(mark + " ");
        }
        System.out.println("n");
    }

    public String toString() {
        return super.toString() + ", RollNo: " + rollno;
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        Student[] students = new Student[3];
        students[0] = new Student(100, "lakhman", 20, new double[]{78, 86, 88, 67, 92});
        students[1] = new Student(101, "shivansh", 21, new double[]{80, 75, 90, 60, 85});
        students[2] = new Student(102, "dev", 22, new double[]{85, 92, 78, 70, 88});

        Student highestAverageStudent = students[0];
```

```
        double highestAverage = students[0].getMarks().length > 0 ?
            calculateAverage(students[0].getMarks()) : 0;

        for (int i = 1; i < students.length; i++) {
            double avg = calculateAverage(students[i].getMarks());
            if (avg > highestAverage) {
                highestAverage = avg;
                highestAverageStudent = students[i];
            }
        }

        System.out.println("Student with the highest average marks:");
        highestAverageStudent.displayDetails();

        Student extraStudent = new Student(103);
        System.out.println(Student.getCount());

        // Close the scanner
        scan.close();
    }

    private static double calculateAverage(double[] marks) {
        double sum = 0;
        for (double mark : marks) {
            sum += mark;
        }
        return marks.length > 0 ? sum / marks.length : 0;
    }
}
```

Output:

LAB_4

Q.1

1. Write a program that catches the divide-by-zero exception using the try-catch mechanism. Take a numeric value and perform division by zero. Catch the ArithmeticException.

Ans.

```
import java.util.Scanner;

public class DivideByZeroException {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter a number: ");

      try {
         int number = scanner.nextInt();
         int result = number / 0;
         System.out.println("Result: " + result);
      } catch (ArithmeticException e) {
         System.out.println("ArithmeticException caught: Division by zero not allowed!");
      }
      scanner.close();
   }
}
```

Output:

java -cp /tmp/Idkbq5Vo27 DivideByZeroException
Enter a number: 20
ArithmeticException caught: Division by zero not allowed!

Q.2

Write a java program using multiple catch blocks. Create a class CatchExercise, inside the try block declare an array a[] with size of 5 elements and initialize with value a[5] =30/5 . Using Multiple catch blocks handle ArithmeticException and ArrayIndexOutOfBoundsException.

Ans.

```
public class CatchExercise {
  public static void main(String[] args) {
    try {
      int[] a = new int[5];
      a[5] = 30 / 7645;
    } catch (ArithmeticException e) {
      System.out.println("ArithmeticException caught: Division by zero not allowed!");
    } catch (ArrayIndexOutOfBoundsException e) {
      System.out.println("ArrayIndexOutOfBoundsException caught: Array index out of range!");
    }
  }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community

ArrayIndexOutOfBoundsException caught: Array index out of range!

Process finished with exit code 0

ArithmeticException caught: Division by zero not allowed!

Process finished with exit code 0

Q.3

Write a program that demonstrates use of finally block. Observe the output of your program for different cases as mentioned below

[U+25CF] Case A: exception does not occur. Perform 25/5 mathematical operation. Catch the NullPointerException.

[U+25CF] Case B: exception occurs but not handled. Perform 25/0 mathematical operation. Catch NullPointerException.

[U+25CF] Case C: exception occurs and handled. Perform 25/0 mathematical operation. Catch ArithmeticException

Ans.

```
public class FinallyBlockDemo {
  public static void main(String[] args) {
    // Case A: No exception
    try {
      int result = 25 / 5;
      System.out.println("Result: " + result);
    } catch (NullPointerException e) {
      System.out.println("NullPointerException caught!");
    } finally {
      System.out.println("Case A: Finally block executed.");
    }

    // Case B: Exception occurs but not handled
    try {
      int result = 25 / 0; // This will throw an ArithmeticException
      System.out.println("Result: " + result);
    } catch (NullPointerException e) {
      System.out.println("NullPointerException caught!");
    } finally {
      System.out.println("Case B: Finally block executed.");
    }

    // Case C: Exception occurs and handled
    try {
      int result = 25 / 0; // This will throw an ArithmeticException
      System.out.println("Result: " + result);
    } catch (ArithmeticException e) {
```

```
        System.out.println("ArithmeticException caught: Division by zero not allowed!");
      } finally {
        System.out.println("Case C: Finally block executed.");
      }
    }
}
```

Output:

Result: 5
Case A: Finally block executed.
Case B: Finally block executed.
Exception in thread "main" java.lang.ArithmeticException: / by zero
at lab4.FinallyBlockDemo.main(FinallyBlockDemo.java:17)

Process finished with exit code 1

Q.4

Create an interface Account with two methods: deposit and withdraw. Create class SavingsAccount which implements the interface. Write a custom Exception handler class CustomException for SavingsAccount to handle the scenarios when the withdrawn amount is larger than the balance in the account.

Ans.

```
interface Account {
    void deposit(double amount);
    void withdraw(double amount) throws InsufficientFundsException;
}

class InsufficientFundsException extends Exception {
    public InsufficientFundsException(String message) {
        super(message);
```

```
    }
}

class SavingsAccount implements Account {
    private double balance;

    @Override
    public void deposit(double amount) {
        balance += amount;
        System.out.println("Deposited: " + amount);
    }

    @Override
    public void withdraw(double amount) throws InsufficientFundsException {
        if (amount > balance) {
            throw new InsufficientFundsException("Insufficient funds!");
        } else {
            balance -= amount;
            System.out.println("Withdrawn: " + amount);
        }
    }

    public double getBalance() {
        return balance;
    }
}

public class TestSavingsAccount {
    public static void main(String[] args) {
        SavingsAccount account = new SavingsAccount();
        account.deposit(1000);

        try {
            account.withdraw(1200); // Withdraw an amount greater than the balance
        } catch (InsufficientFundsException e) {
            System.out.println("InsufficientFundsException caught: " + e.getMessage());
        }
```

```
    }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=57169:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/javabasics/out/production/java basics lab4.TestSavingsAccount
Deposited: 1000.0
InsufficientFundsException caught: Insufficient funds!

Process finished with exit code 0

Q.5

Amethod named add() accepts an array of strings as argument. It converts these to double values and returns their sum. The method generates a NumberFormatException if the element is incorrectly formatted. It can also create and throw a custom exception, RangeException, if an element is less than 0 or greater than 1. Write a program that illustrates how to declare and use this method. Invoke the method from a main. Also provide the finally clause to thank the user for using the program

Ans.

```
import java.util.*;
class RangeException extends Exception {
    public RangeException(String message) {
        super(message);
    }
}
```

```
class Main {
  static double add(String[] values) throws NumberFormatException, RangeException {
    double sum = 0.0;

    try {
      for (String value : values) {
        double num = Double.parseDouble(value);

        if (num < 0 || num > 1) {
          throw new RangeException("Value out of range (0 to 1): " + num);
        }

        sum += num;
      }
    } catch (NumberFormatException e) {
      throw new NumberFormatException("Incorrectly formatted element!");
    }

    return sum;
  }

  public static void main(String[] args) {
    String[] inputs = {"0.5", "0.3", "a", "0.2"};

    try {
      double result = add(inputs);
      System.out.println("Sum of elements: " + result);
    } catch (NumberFormatException | RangeException e) {
      System.out.println("Exception caught: " + e.getMessage());
    } finally {
      System.out.println("Thank you for using the program!");
    }
  }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community
Exception caught: Incorrectly formatted element!
Thank you for using the program!

Process finished with exit code 0

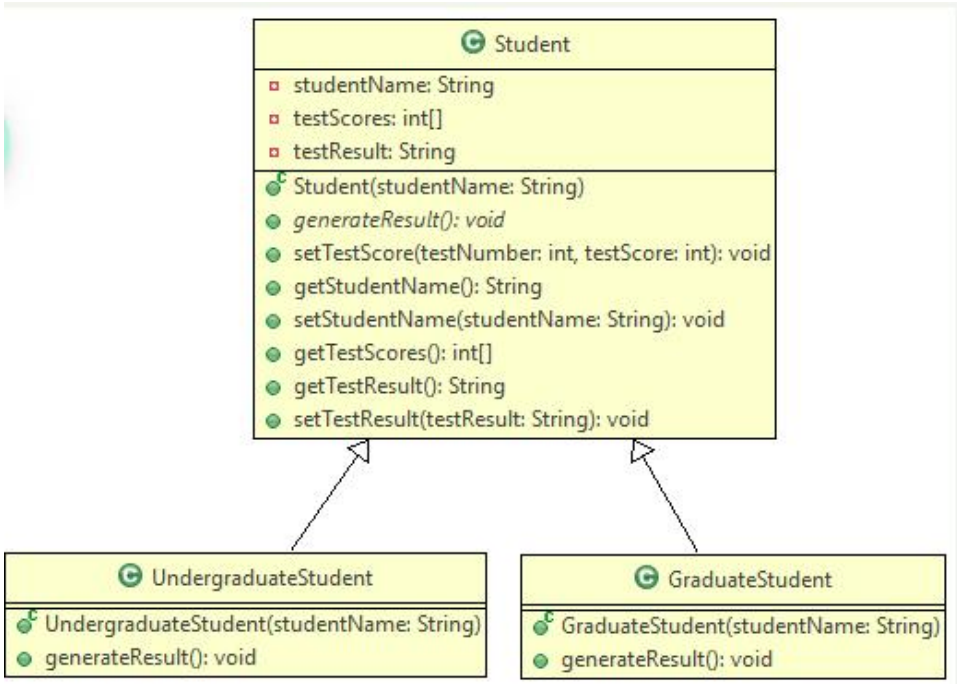
Lab-5

Q.1

LAB 5

Topics :- Abstract class , Interface , Multithreading

1. Anchor College offers both UnderGraduate and PostGraduate programs. The college stores the names of the students, their test scores and the final result for each student. Each student has to take 4 tests in total. You need to create an application for the college by implementing the classes based on the class diagram and description given below



Method Description

Implement the getterand setter methods appr opriately.

1. **Student(Class)**
2. **Student(String studentName)**

- Initialize the instance variable studentName with the value passed to the constructor and other instance variables to the default values.

1. **setTestScore(int testNumber, int testScore)**

- Set the value of the testScore in the appropriate position in the testScores array based on the testNumber

1. **UndergraduateStudent(Class)**
2. **UndergraduateStudent(String studentName)**

- Initialize the instance variable studentName with the value passed to the constructor and other instance variables to the default values.

1. **generateResult()**

- **Implement the abstract method of Student class by setting the value of testResult based on the below details.**

Average Score	**Result**

>=60	Pass
<60	Fail

Sample Input and Output ForUndergraduateStudent

Input:-

Instance Variable	Values
name	**Jerry**
testScores	**{70,69,71,55}**

Output:-

Student Name : Jerry

Result : Pass

1. **PostGraduateStudent(Class)**
2. **PostgraduateStudent(String studentName)**

- Initialize the instance variable studentName with the value passed to the constructor and other instance variables to the default values.

1. **generateResult()**

- **Implement the abstract method of Student class by setting the value of testResult based on the below details.**

Average Score	Result
>=75	Pass
<75	Fail

Sample Input and Output ForPostUndergraduateStudent

Input:-

Instance Variable	Values
name	Tom
testScores	{70,75,80,85}

Output:- Student Name : Tom

Result :

Pass

Ans.

Student.java

```
package lab_5;

abstract class Student {
private String StudentName;
private int[] testScores;
private String testResult;



public Student(String StudentName) {
	super();
	this.StudentName = StudentName;
}

public void setTestScore(int testNumber, int testScore){
	int length=testScores.length;
	testScores[length]=testNumber;

}



public String getStudentName() {
	return StudentName;
}

public void setStudentName(String studentName) {
	StudentName = studentName;
}
```

```
public int[] getTestScores() {
    return testScores;
}

public void setTestScores(int[] testScores) {
    this.testScores = testScores;
}

public String getTestResult() {
    return testResult;
}

public void setTestResult(String testResult) {
    this.testResult = testResult;
}

abstract void generateResult();

}
```

UnderGraduateStudent.java

```
package lab_5;

public class UnderGraduateStudnet extends Student{

    public UnderGraduateStudnet(String name) {
        super(name);
        //TODO Auto-generated constructor stub
    }
    @Override
    void generateResult() {

        System.out.println("Student Name : "+ super.getStudentName());
        int totalScore=0;
        for(int marks:super.getTestScores()) {
```

```
            totalScore+=marks;
        }
//      System.out.println(totalScore);
        int getStatus=totalScore/super.getTestScores().length;
//      System.out.println(getStatus);
        if(getStatus >= 60) {
            super.setTestResult("PASS");
            System.out.println("Result : PASS");
        }
        else {
            super.setTestResult("FAIL");
            System.out.println("Result : FAIL");
        }

    }


}
```

GraduateStudent.java

```
package lab_5;

public class GraduateStudent extends Student {

    public GraduateStudent(String studnetName) {
        super(studnetName);
        //TODO Auto-generated constructor stub
    }

    @Override
    void generateResult() {
        System.out.println("Student Name : "+ super.getStudentName());
        int totalScore=0;
        for(int marks:super.getTestScores()) {
            totalScore+=marks;
```

```
            }
//          System.out.println(totalScore);
            int getStatus=totalScore/super.getTestScores().length;
//          System.out.println(getStatus);
            if(getStatus >= 75) {
                super.setTestResult("PASS");
                System.out.println("Result : PASS");
            }
            else {
                super.setTestResult("FAIL");
                System.out.println("Result : FAIL");
            }
        }


}
```

Main.java

```
package lab_5;

public class Main {
    public static void main(String[] args) {
        UnderGraduateStudnet obj1=new UnderGraduateStudnet("Jerry");
        int[] score={70,69,71,55};
        obj1.setTestScores(score);
        obj1.generateResult();

        GraduateStudent obj2=new GraduateStudent("Tom");
        int[] score2={70,75,80,85};
        obj2.setTestScores(score2);
        obj2.generateResult();

    }
}
```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=49825:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/submission/out/production/submission lab5.lab_5.Main
Student Name : Lakhman
Result : PASS
Student Name : Dev
Result : PASS

Process finished with exit code 0

Q.2

Write a Java program as per the given description to demonstrate use of interface.

1. Define an interface **RelationInterface**.

Write three abstract methods: isGreater, isLess and isEqual.

All methods have a return type of boolean and take an argument of type Line with which the caller object will be compared.

1. Define the **Line** class implements the RelationInterface interface.

- It has 4 double variables for the x and y coordinates of the line.
- Define a **constructor** in **Line** class that initializes these 4 variables.
- Define a method **getLength()** that computes length of the line.

[double length = Math.sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1))].

- Implement the methods of interface in Line class

1. In class **CompareLines**.Java, create two objects of Line class, call the three methods to compare the lengths of the lines.

Ans.

```
package lab_5;

public interface RelationInterface {
   boolean isGreater(Line otherLine);
   boolean isLess(Line otherLine);
   boolean isEqual(Line otherLine);
}
```

Line.java

```
package lab_5;



public class Line implements RelationInterface {
   private double x1, y1, x2, y2;

   public Line(double x1, double y1, double x2, double y2) {
      this.x1 = x1;
      this.y1 = y1;
      this.x2 = x2;
      this.y2 = y2;
   }

   public double getLength() {
      return Math.sqrt((x2 - x1) * (x2 - x1) + (y2 - y1) * (y2 - y1));
   }

   @Override
   public boolean isGreater(Line otherLine) {
      return this.getLength() > otherLine.getLength();
   }

   @Override
   public boolean isLess(Line otherLine) {
      return this.getLength() < otherLine.getLength();
```

```
    }

    @Override
    public boolean isEqual(Line otherLine) {
        return this.getLength() == otherLine.getLength();
    }
}
```

```
/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -
javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community
Edition.app/Contents/lib/idea_rt.jar=49836:/Users/lakhman/Applications/IntelliJ IDEA Community
Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -
Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4
coding/java/submission/out/production/submission lab5.lab_2.CompareLines
Is line1 greater than line2? false
Is line1 less than line2? true
Is line1 equal to line2? false

Process finished with exit code 0
```

CompareLines.java

```
package lab_5;

public class CompareLines {
    public static void main(String[] args) {

        Line line1 = new Line(0, 0, 3, 4);
        Line line2 = new Line(0, 0, 5, 12);


        System.out.println("Is line1 greater than line2? " + line1.isGreater(line2));
        System.out.println("Is line1 less than line2? " + line1.isLess(line2));
        System.out.println("Is line1 equal to line2? " + line1.isEqual(line2));
    }
}
```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=49836:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/submission/out/production/submission lab5.lab_2.CompareLines

Is line1 greater than line2? false
Is line1 less than line2? true
Is line1 equal to line2? false

Process finished with exit code 0

Q.3

In the producer–consumer problem, the producer and the consumer share a common, fixed-size buffer used as a queue. (Let buffer size as 1). The producer's job is to generate data, put it into the buffer. At the same time, the consumer is consuming the data (i.e. removing it from the buffer). The problem is to make sure that the producer won't try to add data into the buffer if it's full and that the consumer won't try to remove data from an empty buffer. Write a Java application consisting of all necessary classes to achieve this.

Ans.

Buffer.java

```
public class Buffer {
   private int data;
   private boolean empty;
```

```
    public Bufer() {
        empty = true;
    }

    public synchronized void produce(int newData) {
        while (!empty) {
            try {
                wait();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
        data = newData;
        empty = false;
        System.out.println("Produced: " + data);
        notify();
    }

    public synchronized int consume() {
        while (empty) {
            try {
                wait();
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
        System.out.println("Consumed: " + data);
        empty = true;
        notify();
        return data;
    }
}
```

Producer.java

```
import java.util.Random;

public class Producer implements Runnable {
    private Buffer buffer;
    private Random random;

    public Producer(Buffer buffer) {
        this.buffer = buffer;
        this.random = new Random();
    }

    @Override
    public void run() {
        while (true) {
            int newData = random.nextInt(100); // Generate random data
            buffer.produce(newData); // Put data into the buffer
            try {
                Thread.sleep(random.nextInt(3000)); // Sleep for random time
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
    }
}
```

Consumer.java

```
import java.util.Random;

public class Consumer implements Runnable {
```

```
    private Buffer buffer;

    public Consumer(Buffer buffer) {
        this.buffer = buffer;
    }

    @Override
    public void run() {
        while (true) {
            buffer.consume(); // Consume data from the buffer
            try {
                Thread.sleep(new Random().nextInt(3000)); // Sleep for random time
            } catch (InterruptedException e) {
                Thread.currentThread().interrupt();
            }
        }
    }
}
```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=49942:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/submission/out/production/submission lab5.que_3.Main
Produced: 58
Consumed: 58
Produced: 41
Consumed: 41
Produced: 12
Consumed: 12
Produced: 35
Consumed: 35
Produced: 0
Consumed: 0

Produced: 69
Consumed: 69
Produced: 82
Consumed: 82
Produced: 40
Consumed: 40
Produced: 27
Consumed: 27
Produced: 96
Consumed: 96
Produced: 67
Consumed: 67
Produced: 38
Consumed: 38
Produced: 4
Consumed: 4
Produced: 9
Consumed: 9
Produced: 48
Consumed: 48
Produced: 24
Consumed: 24
Produced: 72
Consumed: 72
Produced: 1
Consumed: 1
Produced: 19
Consumed: 19
Produced: 38
Consumed: 38
Produced: 34
Consumed: 34

Process finished with exit code 130 (interrupted by signal 2: SIGINT)

Q.4

Write a multithreaded Java application to produce a deadlock condition.

Ans.

```
public class DeadlockExample {
  private static final Object lock1 = new Object();
  private static final Object lock2 = new Object();

  public static void main(String[] args) {
    Thread thread1 = new Thread(() -> {
      synchronized (lock1) {
        System.out.println("Thread 1 acquired lock1");
        try {
          Thread.sleep(100); // Introducing delay to make deadlock more apparent
        } catch (InterruptedException e) {
          e.printStackTrace();
        }
        synchronized (lock2) {
          System.out.println("Thread 1 acquired lock2");
        }
      }
    });

    Thread thread2 = new Thread(() -> {
      synchronized (lock2) {
        System.out.println("Thread 2 acquired lock2");
        synchronized (lock1) {
          System.out.println("Thread 2 acquired lock1");
        }
      }
    });

    thread1.start();
    thread2.start();
  }
```

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=49920:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/submission/out/production/submission lab5.DeadlockExample
Thread 1 acquired lock1
Thread 2 acquired lock2

LAB-6 JDBC, Generics

Q.1

Write a Java application to perform operations for student information like (id[Primary key, Auto increment],firstName, lastName, branch, username and password) from a database using JDBC.

Ans.

Insert two records for student

```
public class Main {

    public static void main(String[] args) throws SQLException {
        // JDBC URL, username, and password of MySQL server
        String url = "jdbc:mysql://localhost:3306/lucky";
        String user = "root";
        String password = "";

        try {
```

```
        // Establish a connection
        Connection connection = DriverManager.getConnection(url, user, password);
        System.out.println("Connected to the database!");

        // Create a statement
        Statement statement = connection.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE,
ResultSet.CONCUR_READ_ONLY);


        String query1 = "INSERT INTO students (firstName, lastName, branch, username, password)
VALUES (?, ?, ?, ?, ?)";
       PreparedStatement preparedStatement1 = connection.prepareStatement(query1);
          preparedStatement1.setString(1, "parmar");
          preparedStatement1.setString(2, "lakhman");
          preparedStatement1.setString(3, "ce");
          preparedStatement1.setString(4, "lucky737");
          preparedStatement1.setString(5, "l@123");
          preparedStatement1.executeUpdate();
          System.out.println("Student 1 inserted successfully");

        preparedStatement1.setString(1, "devin");
        preparedStatement1.setString(2, "patel");
        preparedStatement1.setString(3, "ce");
        preparedStatement1.setString(4, "dev737");
        preparedStatement1.setString(5, "d@123");
        preparedStatement1.executeUpdate();
        System.out.println("Student 2 inserted successfully");
        // Execute a query

        String slectquery="select * from students";

//
        ResultSet resultSet = statement.executeQuery(slectquery);
// resultSet.next();
        //use of absolute
        resultSet.absolute(1); // Move to the 5th row
        System.out.println("Cursor moved to 2nd row: " + getRowData(resultSet));
        resultSet.relative(0);
```

```
        System.out.println("Cursor moved to 3rd row: " + getRowData(resultSet));

        //isfirtst islast to see cursor
        // Check Cursor Position
        System.out.println("Is cursor at the first row? " + resultSet.isFirst());
        System.out.println("Is cursor at the last row? " + resultSet.isLast());

        // Navigate to Extremes
        resultSet.first(); // Move to the first row
        System.out.println("Cursor moved to the first row: " + getRowData(resultSet));
        resultSet.last(); // Move to the last row
        System.out.println("Cursor moved to the last row: " + getRowData(resultSet));

        resultSet.afterLast();
        while (resultSet.previous()){
          System.out.println(getRowData(resultSet));
        }



        resultSet.close();
        statement.close();
      connection.close();
      } catch (SQLException e) {
        e.printStackTrace();
      }
    }

    private static String getRowData(ResultSet resultSet) throws SQLException {

            int id = resultSet.getInt("id");
            String firstnamename = resultSet.getString("firstname");
            String lastname=resultSet.getString("lastname");
            String branch=resultSet.getString("branch");
            String username=resultSet.getString("username");
            String pass=resultSet.getString("password");

            //Add more columns as needed
           return "id : "+ id +", name : "+firstnamename + " " + lastname +", branch :" +
```

```
                branch+ ", username : "+username+", pass : "+pass;
// // Print the results
  }
}
```

output:

Connected to the database!
Student 1 inserted successfully
Student 2 inserted successfully
Cursor moved to 2nd row: id : 1, name : parmar lakhman, branch :ce, username : lucky737, pass : l@123
Cursor moved to 3rd row: id : 1, name : parmar lakhman, branch :ce, username : lucky737, pass : l@123
Is cursor at the first row? true
Is cursor at the last row? false
Cursor moved to the first row: id : 1, name : parmar lakhman, branch :ce, username : lucky737, pass : l@123
Cursor moved to the last row: id : 2, name : devin patel, branch :ce, username : dev737, pass : d@123
id : 2, name : devin patel, branch :ce, username : dev737, pass : d@123
id : 1, name : parmar lakhman, branch :ce, username : lucky737, pass : l@123

Q.2

Using JDBCAPI and MySql database perform the following operations.

create a table MOVIES with following columns in the database:

Id of type**INTEGER AUTO INCREMENT**,

Title of type**VARCHAR (50)**,

Genreof type **VARCHAR (50)**,

YearOfReleaseof type **INTEGER**.

Define **Movie**class with following data members

private Integer id;

private String title;

private String genre;

private Integer yearOfRelease;

Create getters and setters for the mentioned data members.

1. Define following methods in a class, test the methods according to user input
2. **createMovie()**it will insert a new record for a movie
3. **deleteMovie(int MovieID)**it will delete a movie with given MovieID
4. **updateMovieTitle(String title, int id)**it will update the title of a movie with given id.
5. **findMovieById(int MovieId)**it will display all details of a movie with a given MovieId
6. **findAllMovie()**it will display all details of all movies

Ans.

```
import java.sql.*;

class Movies{


  private int id;
  private String title;
   private String genre;
   private int yearOfRelease;
```

```
public Movies(int id, String title, String genre, int yearOfRelease) {
   this.id = id;
   this.title = title;
   this.genre = genre;
   this.yearOfRelease = yearOfRelease;
}

public Movies() {

}

public int getId() {
   return id;
}

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

public String getTitle() {
   return title;
}

public void setTitle(String title) {
   this.title = title;
}

public String getGenre() {
   return genre;
}

public void setGenre(String genre) {
   this.genre = genre;
}

public int getyearOfRelease() {
   return yearOfRelease;
}
```

```
    public void setyearOfRelease(int yearOfRelease) {
        this.yearOfRelease = yearOfRelease;
    }

    boolean createMovie(Connection connection) throws SQLException {
        String inserter="INSERT INTO `MOVIES`(`id`, `Title`, `Genre`, `YearOfRelease`) VALUES
(NULL, '"+this.title+"', '"+this.genre+"', '"+this.getyearOfRelease()+"');";
        Statement statement=connection.createStatement();


        return statement.executeUpdate(inserter)==1;
    }
    int deleteMovie(int MovieID,Connection connection) throws SQLException {

// DELETE FROM `MOVIES` WHERE `MOVIES`.`id` = 1;
        String inserter="DELETE FROM `MOVIES` WHERE `MOVIES`.`id` = "+MovieID+";";
        Statement statement=connection.createStatement();
        return statement.executeUpdate(inserter);


    }
    boolean updateMovieTitle(String title,int MovieID,Connection connection) throws SQLException {
// UPDATE `MOVIES` SET `Genre` = 'comedy ,reallife' WHERE `MOVIES`.`id` = 2;
        String updater="UPDATE `MOVIES` SET `Title` = '"+title+"' WHERE `MOVIES`.`id` =
"+MovieID+";";
        Statement statement=connection.createStatement();
        return statement.execute(updater);
    }
    ResultSet findMovieById(int MovieId ,Connection connection) throws SQLException {
        String finder="SELECT * FROM `MOVIES` WHERE `id` = "+MovieId+" ";
        Statement statement=connection.createStatement();
        ResultSet resultSet = statement.executeQuery(finder);


        return resultSet;
    }
    ResultSet findAllMovie(Connection connection) throws SQLException {
        String finder="SELECT * FROM `MOVIES`";
        Statement statement=connection.createStatement();
```

```
        ResultSet resultSet = statement.executeQuery(finder);

        return resultSet;
    }
    void printallmovie(ResultSet resultSet) throws SQLException {
        resultSet.next();
        while (resultSet.next()) {
            int id=resultSet.getInt("id");
            String title=resultSet.getString("Title");

             String genre=resultSet.getString("Genre");
            int yearOfRelease=resultSet.getInt("YearOfRelease");
           System.out.println("id: "+id+", Title: "+title+", Genre: "+genre+", YearOfRelease:
"+yearOfRelease);
        }

    }
    void printmovie(ResultSet resultSet) throws SQLException {
        resultSet.next();

            int id=resultSet.getInt("id");
            String title=resultSet.getString("Title");

            String genre=resultSet.getString("Genre");
            int yearOfRelease=resultSet.getInt("YearOfRelease");
            System.out.println("id: "+id+", Title: "+title+", Genre: "+genre+", YearOfRelease:
"+yearOfRelease);



    }
}

public class Movie {

    public static void main(String[] args) {
        String url = "jdbc:mysql://localhost:3306/lucky";
        String user = "root";
        String password = "";
```

```
try {
    Connection connection = DriverManager.getConnection(url, user, password);
    System.out.println("Connected to the database");

    Movies movieHandler = new Movies();

    // Insert a new movie
    Movies newMovie = new Movies(10, "Avengers", "Sci-Fi", 2014);
    boolean insertionResult = newMovie.createMovie(connection);
    System.out.println("Movie Inserted: " + insertionResult);

    // Update the title of a movie
    boolean updateResult = movieHandler.updateMovieTitle("Avengers Endgame", 10, connection);
    System.out.println("Movie Title Updated: " + updateResult);

    // Display details of a movie by ID
    ResultSet movieById = movieHandler.findMovieById(4, connection);
    System.out.println("Details of Movie with ID 4:");
    movieHandler.printmovie(movieById);

    // Display details of all movies
    ResultSet allMovies = movieHandler.findAllMovie(connection);
    System.out.println("\nDetails of All Movies:");
    movieHandler.printallmovie(allMovies);

    // Delete a movie by ID
    int movieIdToDelete = 4;
    int deleteResult = movieHandler.deleteMovie(movieIdToDelete, connection);
    System.out.println("\nMovie Deleted: " + (deleteResult > 0));

    // Display details of all movies after deletion
    ResultSet remainingMovies = movieHandler.findAllMovie(connection);
    System.out.println("\nDetails of Remaining Movies:");
    movieHandler.printallmovie(remainingMovies);

} catch (SQLException e) {
    throw new RuntimeException(e);
```

```
    }
  }
}
```

output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=61367:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/jdbc/out/production/jdbc:/Users/lakhman/Downloads/mysql-connector-j-8.2.0/mysql-connector-j-8.2.0.jar Movie
Connected to the database
Movie Inserted: true
Movie Title Updated: false
Details of Movie with ID 4:
id: 4, Title: IRON MAN, Genre: scifi, YearOfRelease: 2008

Details of All Movies:
id: 3, Title: ironman2, Genre: ironman3, YearOfRelease: 2008
id: 4, Title: IRON MAN, Genre: scifi, YearOfRelease: 2008
id: 7, Title: Avengers, Genre: Sci-Fi, YearOfRelease: 2014

Movie Deleted: true

Details of Remaining Movies:
id: 3, Title: ironman2, Genre: ironman3, YearOfRelease: 2008
id: 7, Title: Avengers, Genre: Sci-Fi, YearOfRelease: 2014

Process finished with exit code 0

Q.3

1. Create a Generic class Calculator which can perform addition, subtraction, multiplication and division. Make sure that Calculator class works for Numeric values only. Write an appropriate main method in TestCalculator class.

Ans.

```
class calc<T extends Number>{
   T op1;
   T op2;

   public calc(T op1, T op2) {
      this.op1 = op1;
      this.op2 = op2;
   }

   public calc() {

   }

   double addition(T op1, T op2) {
      return op1.doubleValue()+op2.doubleValue();

   }
   double subtraction(T op1, T op2) {
      return op1.doubleValue()- op2.doubleValue();
   }
   double multiplication(T op1, T op2) {
      return op1.doubleValue()*op2.doubleValue();
   }
   double division(T op1, T op2) {
      return op1.doubleValue()/op2.doubleValue();
   }
```

```
}
public classTestCalculator {

   public static void main(String[]gs) {

      calc c1=new calc();
      c1.op1=34234;
      c1.op2=234.3545;
      System.out.println(c1.addition(354,3453.354342));
      System.out.println(c1.subtraction(c1.op1, c1.op2));
      System.out.println(c1.multiplication(c1.op1, c1.op2));
      System.out.println(c1.division(c1.op1, c1.op2));
   }
}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=61454:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/jdbc/out/production/jdbc:/Users/lakhman/Downloads/mysql-connector-j-8.2.0/mysql-connector-j-8.2.0.jar TestCalculator
3807.354342
33999.6455
8022891.953
146.07784360872097

Process finished with exit code 0

Q.4

1. Write a Java program to create a generic method that takes two arrays of T type and checks if they have the same elements in the same order.

Ans.

```
public class check {
   public static <T> boolean arrEquals(T[] arr1, T[] arr2){
      if (arr1.length != arr2.length) return false;
      for(int i=0;i<arr1.length;i++) {
         if (!arr1[i].equals(arr2[i])) return false;
      }
      return true;
   }
   public static void main(String[] args) {
      Integer[] intArray1 = {1, 2, 3, 4, 5};
      Integer[] intArray2 = {1, 2, 3, 4, 5};

      boolean intArraysEqual = arrEquals(intArray1, intArray2);
      System.out.println("Integer Arrays are equal: " + intArraysEqual);

      // Example with String arrays
      String[] strArray1 = {"apple", "orange", "banana"};
      String[] strArray2 = {"apple", "orange", "banana"};

      boolean strArraysEqual = arrEquals(strArray1, strArray2);
      System.out.println("String Arrays are equal: " + strArraysEqual);

      // Example with Double arrays
      Double[] doubleArray1 = {1.0, 2.0, 3.1};
      Double[] doubleArray2 = {1.0, 2.0, 3.0};

      boolean doubleArraysEqual = arrEquals(doubleArray1, doubleArray2);
      System.out.println("Double Arrays are equal: " + doubleArraysEqual);
   }

}
```

Output:

/Library/Java/JavaVirtualMachines/jdk-21.jdk/Contents/Home/bin/java -javaagent:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/lib/idea_rt.jar=61632:/Users/lakhman/Applications/IntelliJ IDEA Community Edition.app/Contents/bin -Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding=UTF-8 -classpath /Users/lakhman/Desktop/sem 4 coding/java/jdbc/out/production/jdbc:/Users/lakhman/Downloads/mysql-connector-j-8.2.0/mysql-connector-j-8.2.0.jar check
Integer Arrays are equal: true
String Arrays are equal: true
Double Arrays are equal: false

Process finished with exit code 0

Lab 7 Collection , I/O

Q.1

Write a Java program that accepts two filenames. Based on the user's choice to copy or append, copy the first file into the second file or append the content of the first file to the second file.

Ans.

package lab7.que1;

```
import java.io.FileInputStream;
import java.io.FileOutputStream;
import java.io.IOException;
import java.util.Scanner;

public class InputOutput {
   public static void main(String[] args) {
      Scanner scanner=new Scanner(System.in);
      System.out.println("enter first file name");
      String st1=scanner.nextLine();
      System.out.println("enter second file name");
      String st2=scanner.nextLine();
      System.out.println("enter 0 for copying "+ st1 +" to" + st2 +"1 appending ");
      boolean flag=scanner.nextBoolean();
      StringBuffer st=new StringBuffer("/Users/lakhman/Desktop/sem 4
coding/java/submission/src/lab7/que1/");
      String filePath = String.valueOf(st.append(st1));

      String filepath2=String.valueOf(st.append(st2));
      try {
         FileInputStream in = new FileInputStream(filePath);
         FileOutputStream out=null;
         if(flag) out = new FileOutputStream(filepath2);
         else out = new FileOutputStream(filepath2,true);

         int temp;


            while ((temp = in.read()) != -1) {


               out.write(temp);

         }
      } catch (IOException e) {

         e.printStackTrace();
      }
```

```
    }
}
```

Output:

```
enter first file name
F1.txt
enter second file name
lucky.txt
enter 0 for copying F1.txt to lucky.txt1 appending
0

Process finished with exit code 0
```

Q.2

Write a Java program to generate a linked list of some five students (Student objects) and display the list of students in a sorted order as per name of students.

Ans.

```
package lab7.que1;
import java.util.*;
class Student {
    int rollno;
    String name;




}
class StudentComparator implements Comparator<Student> {
    @Override
    public int compare(Student student1, Student student2) {
        // Compare students based on their names
        return student1.name.compareTo(student2.name);
    }
```

```
}
public class Linklistquestion {
   public static void main(String[]gs) {
      Student st=new Student();
      LinkedList<Student> ll = new LinkedList<Student>();
      st.name="lakhman";
      st.rollno=30;
      Student st2=new Student();
      st2.rollno=33;
      st2.name="vaibha";
      Student st3=new Student();
      st3.rollno=3;
      st3.name="shivansh";
      Student st4=new Student();
      st4.rollno=13;
      st4.name="pradip";
      Student st5=new Student();
      st5.rollno=43;
      st5.name="raza";
      ll.add(st);
      ll.add(st2);
      ll.add(st3);
      ll.add(st4);
      ll.add(st5);
// ll.sort(Comparator .comparingInt(student -> student.rollno));
// ll.sort(Comparator .comparing(studnet -> studnet.name));
      ll.sort(new StudentComparator());
      for(Student studnet:ll) {
         System.out.println(studnet.rollno+" "+studnet.name);
      }
   }
}
```

Output:

```
lakhman 30
pradip 13
```

raza 43
shivansh 3
vaibha 33

Process finished with exit code 0

Q.3

1. Write a Java program to map Person objects to string hobby using TreeMap class. This mapping should store Person objects in ascending sorting by their name.

Persons	hobby
Name: Bhairav Age: 22	Singing
Name: Dhara Age:23	Sketching
Name:Anmol Age: 23	Reading
Name: Megh Age 21	Singing
Name: Raag Age:22	Sketching

Find the unique list/set of all the hobbies that are mapped in this collection and display it.

Ans.

```
package lab7.que1;

import java.util.Arrays;
import java.util.HashMap;
import java.util.*;

class Person implements Comparable<Person>{
   int age;
   String name;

   public Person(int age, String name) {
      this.age = age;
      this.name = name;
   }
```

```
@Override public int compareTo(Person p1) {
    return this.name.compareTo(p1.name);
  }
}
public class Treemaps {
  public static void main(String[] args) {
    String[] hobby={"Singing", "Sketching", "Reading","Singing","Sketching"};
    Person person1=new Person(22,"bhavya");
    Person person2=new Person(15,"nihar");
    Person person3=new Person(28,"duryodhan");
    Person person4=new Person(10,"karn");
    Person person5=new Person(50,"parth");
    Map<Person,String> newmap= new TreeMap<Person,String>();

    newmap.put(person1,hobby[0]);
    newmap.put(person2,hobby[1]);
    newmap.put(person3,hobby[2]);
    newmap.put(person4,hobby[3]);
    newmap.put(person5,hobby[4]);
    Set<String> uniuqe=new HashSet<>();
    uniuqe.addAll(List.of(hobby));

    System.out.println();
    for (Person p:newmap.keySet()){
      System.out.println(p.name +" "+ p.age);
    }

    for(String lists:uniuqe) {
      System.out.println(lists);
    }









  }
}
```

Output:

```
Reading
Singing
Sketching
bhargav duryodhan karn
10
nihar
15
parth
50
22
28
Process finished with exit code 0
```

Q.4

Write a generic interface MinMax having two methods findMin() and findMax(). Create a class named MyClass which implements the above interface. Write an appropriate demo class to test your classes and interfaces. Create an object of MyClass which stores an array of Book and find books having minimum and maximum price.

Ans.

```
package lab7.que1;

class MyClass<T extends Comparable<T>> implements MinMax<T> {

    @Override
    public T findMin(T[] arr) {
        if (arr == null || arr.length == 0) {
            return null;
        }
```

```
        T min = arr[0];
        for (T item : arr) {
            if (item.compareTo(min) < 0) {
                min = item;
            }
        }
        return min;
    }

    @Override
    public T findMax(T[] arr) {
        if (arr == null || arr.length == 0) {
            return null;
        }

        T max = arr[0];
        for (T item : arr) {
            if (item.compareTo(max) > 0) {
                max = item;
            }
        }
        return max;
    }
}
class Book implements Comparable<Book> {
    String title;
    double price;

    public Book(String book1, double price) {
        this.title=book1;
        this.price=price;
    }

    // Constructor and other methods for Book class

    @Override
    public int compareTo(Book other) {
        // Implement comparison based on book prices
```

```
        return Double.compare(this.price, other.price);
    }

    public String getTitle() {
        return title;
    }
}
public class Demo {
    public static void main(String[] args) {
        Book[] books = {
                new Book("Book1", 20.5),
                new Book("Book2", 15.8),
                new Book("Book3", 25.3),
                //Add more books as needed
        };

        // Create an object of MyClass
        MyClass<Book> myClass = new MyClass<>();

        // Find the book with the minimum price
        Book minPriceBook = myClass.findMin(books);
        System.out.println("Book with Minimum Price: " + minPriceBook.getTitle());

        // Find the book with the maximum price
        Book maxPriceBook = myClass.findMax(books);
        System.out.println("Book with Maximum Price: " + maxPriceBook.getTitle());
    }
}
```

Output:

```
Book with Minimum Price: Book2
Book with Maximum Price: Book3

Process finished with exit code 0
```

Lab-8 servlets

WelcomeServlet.java

```
//WelcomeServlet.java
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;

public classWelcomeServlet extends HttpServlet {
   public void doGet(HttpServletRequest request, HttpServletResponse response) throws
ServletException, IOException {
      response.setContentType("text/html");
      PrintWriter out = response.getWriter();

      out.println("<html>");
      out.println("<head>");
      out.println("<title>Welcome Message</title>");
      out.println("</head>");
      out.println("<body>");
      out.println("<h1>Welcome to the Web Application!</h1>");
      out.println("</body>");
      out.println("</html>");
   }
}
```

web.xml

```
<!– web.xml –>
<web-app>
   <servlet>
      <servlet-name>WelcomeServlet</servlet-name>
      <servlet-class>WelcomeServlet</servlet-class>
   </servlet>
```

```
    <servlet-mapping>
        <servlet-name>WelcomeServlet</servlet-name>
        <url-pattern>/welcome</url-pattern>
    </servlet-mapping>
</web-app>
```

index.jsp

```
<!– index.jsp –>
<!DOCTYPE html>
<html>
<head>
    <meta charset="UTF-8">
    <title>Lab-8</title>
</head>
<body>
    <h1>Lab-8</h1>
    <p>This is a simple JAVA web application.</p>
    <p><a href="welcome">Click here</a> to view the welcome message.</p>
</body>
</html>
```

Hello World!

Click here to view the welcome message.

Hello World!

Java Web Application - Login Module

index.html

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!– index.html –>
<!DOCTYPE html>
<html>
<head>
  <meta charset="UTF-8">
  <title>Login Page</title>
</head>
<body>

<!– Input fields for username and password –>
<form action="ValidateServlet" method="post">
  <label for="username">Username:</label>
  <input type="text" id="username" name="username" required>


  <label for="password">Password:</label>
  <input type="password" id="password" name="password" required>


  <input type="submit" value="Login">
</form>
</body>
</html>
```

ValidateServlet.java

```
package org.example.lab8;

import java.io.*;

import jakarta.servlet.RequestDispatcher;
import jakarta.servlet.ServletException;
```

```
import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

@WebServlet(name = "ValidateServlet", value = "/ValidateServlet")
public class ValidateServlet extends HttpServlet {
   private String message;

   public void init() {
      message = "Hello World!";
   }


   public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException,
ServletException {
      response.setContentType("text/html");

      String username = request.getParameter("username");
      String password = request.getParameter("password");

      // Check if the username and password match the specified values
      if ("lakhman".equals(username) && "123".equals(password)) {
         // If matched, display a success message
         request.setAttribute("username",username);
         request.getRequestDispatcher("welcome.jsp").forward(request,response);

      } else {
         // If not matched, redirect back to the index.html page
         PrintWriter out = response.getWriter();
         out.println("<html><body>");
         out.println("<h1>Login UnSuccessful!</h1>");
         out.println("</body></html>");
         RequestDispatcher requestDispatcher=request.getRequestDispatcher("index.jsp");
         requestDispatcher.include(request,response);
      }
   }


   public void destroy() {
```

```
    }
}
```

Welcome.jsp

```
<%--
  Created by IntelliJ IDEA.
  User: lakhman
  Date: 19/02/24
  Time: 12:36am
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<html>
<head>
    <title>Title</title>
</head>
<body>
<h1>
    Welcome ${username}
</h1>
</body>
</html>
```

index.html

Username: lakhman Password: ••• Login

Welcome lakhman


localhost:8080/lab8_war_exploded/
Username: pamar
Password:
Login


localhost:8080/lab8_war_exploded/ValidateServlet
Login UnSuccessful!
Username:
Password:
Login

Lab-9

JSP, Servlet, Session Management in web application

Topics:

1. Question:

Write a Java web application using **HttpSession** which allows only logged in users to access the other JSPs/Servlets of the application. Write the following components:

1. **Login.html** allows users to provide username and password and send them as request parameters to LoginVerifierServlet.
2. **LoginVerifierServlet takes username and password from login.html and verifies it. If credentials are correct then it creates a session. It displays welcome message along with username and links to first.jsp and second.jsp.**
3. **first.jsp** and **second.jsp** display some text with username and can be accessed if the user is logged in. (you should delegate to Login.html if the user is not logged in)

Login.html

```
<%@ page contentType="text/html; charset=UTF-8" pageEncoding="UTF-8" %>
<!DOCTYPE html>
<html>
<head>
  <title>Login</title>
</head>
<body>
<h2>Login</h2>
<form action="LoginVerifierServlet" method="post">
  Username: <input type="text" name="username"><br>
```

```
    Password: <input type="password" name="password"><br>
    <input type="submit" value="Login">
</form>
</body>
</html>
```

LoginVerifierServlet

```
package org.lakhman.lab9;

import java.io.*;

import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

@WebServlet(name = "helloServlet", value = "/LoginVerifierServlet")
public class HelloServlet extends HttpServlet {
    private String message;

    public void init() {
        message = "Hello World!";
    }

    public void doPost(HttpServletRequest request, HttpServletResponse response) throws IOException {
        String username = request.getParameter("username");
        String password = request.getParameter("password");

        // For simplicity, hardcoding the correct username and password
        if ("admin".equals(username) && "password".equals(password)) {
            HttpSession session = request.getSession();
            session.setAttribute("username", username);
            response.sendRedirect("WelcomeServlet");
        } else {
            response.sendRedirect("Login.html");
        }
```

```
    }

    public void destroy() {
    }
}
```

WelcomeServlet.java

```
package org.lakhman.lab9;

import java.io.*;

import jakarta.servlet.http.*;
import jakarta.servlet.annotation.*;

@WebServlet(name = "helloServlet", value = "/helloServlet")
public class HelloServlet extends HttpServlet {
    private String message;

    public void init() {
        message = "Hello World!";
    }

    public void doGet(HttpServletRequest request, HttpServletResponse response) throws IOException {
        HttpSession session = request.getSession();
        String username = (String) session.getAttribute("username");
        response.setContentType("text/html");

        if (username != null) {
            response.getWriter().println("<h1>Welcome, " + username + "!</h1>");
            response.getWriter().println("<a href='first.jsp'>First Page</a><br>");
            response.getWriter().println("<a href='second.jsp'>Second Page</a>");
        } else {
            response.sendRedirect("Login.html");
        }
    }
```

```
    public void destroy() {
    }
}
```

First.jsp

```
<%--
  Created by IntelliJ IDEA.
  User: lakhman
  Date: 20/02/24
  Time: 10:24am
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <title>First Page</title>
</head>
<body>
<h1>First Page</h1>
<% String username = (String) session.getAttribute("username"); %>
<% if (username != null) { %>
<p>Welcome, <%= username %>!</p>
<p>This is the first page.</p>
<% } else { %>
<% response.sendRedirect("Login.html"); %>
<% } %>
</body>
</html>
```

Second.jsp

```
<%--
  Created by IntelliJ IDEA.
  User: lakhman
  Date: 20/02/24
  Time: 10:24am
  To change this template use File | Settings | File Templates.
--%>
<%@ page contentType="text/html;charset=UTF-8" language="java" %>
<!DOCTYPE html>
<html>
<head>
    <title>First Page</title>
</head>
<body>
<h1>First Page</h1>
<% String username = (String) session.getAttribute("username"); %>
<% if (username != null) { %>
<p>Welcome, <%= username %>!</p>
<p>This is the first page.</p>
<% } else { %>
<% response.sendRedirect("Login.html"); %>
<% } %>
</body>
</html>
```


localhost:8080/lab9_war_exploded/Login.html
Login
Username: admin
Password:
Login


localhost:8080/lab9_war_exploded/WelcomeServlet
Welcome, admin!
First Page
Second Page

First Page

Welcome, admin!

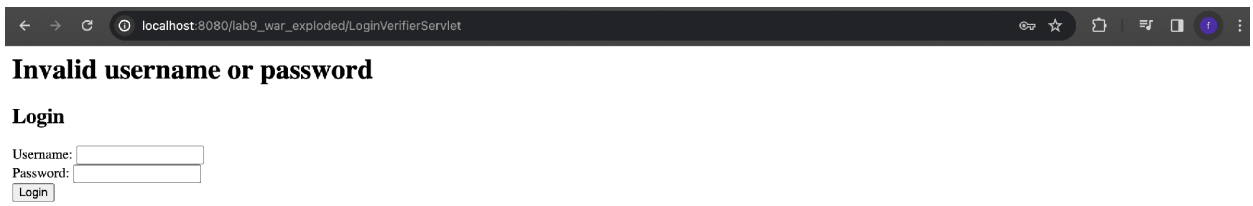
This is the first page.

Second Page

Welcome, admin!

This is the second page.


localhost:8080/lab9_war_exploded/Login.html
Login
Username: lakhman
Password:
Login


localhost:8080/lab9_war_exploded/LoginVerifierServlet
Invalid username or password
Login
Username:
Password:
Login

1. Write a web based java application containing a JSP which performs the simple arithmetic calculation. Take the necessary operands and operators in textboxes. Write your JSP code using jsp:useBean action tag.

CalculatorBean

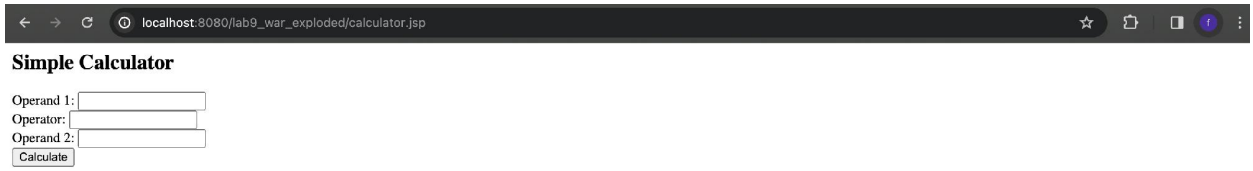
```
package org.lakhman.lab9;

public class CalculatorBean {
   private double operand1;
   private double operand2;
   private String operator;
   private double result;

   public double getOperand1() {
      return operand1;
   }

   public void setOperand1(double operand1) {
      this.operand1 = operand1;
   }

   public double getOperand2() {
      return operand2;
   }

   public void setOperand2(double operand2) {
      this.operand2 = operand2;
   }

   public String getOperator() {
      return operator;
   }
```

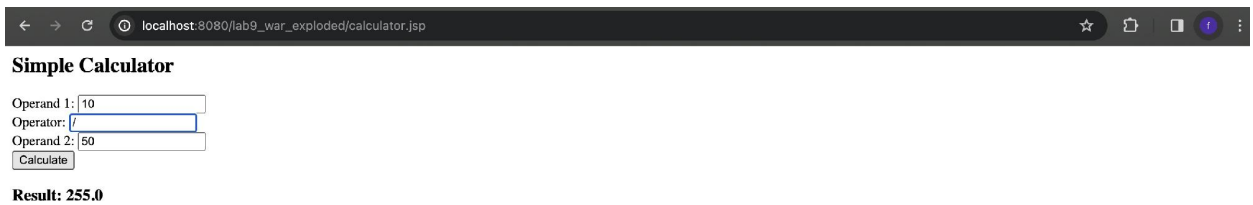
```
  public void setOperator(String operator) {
    this.operator = operator;
  }

  public double getResult() {
    switch (operator) {
      case "+":
        result = operand1 + operand2;
        break;
      case "-":
        result = operand1 - operand2;
        break;
      case "*":
        result = operand1 * operand2;
        break;
      case "/":
        if (operand2 != 0) {
          result = operand1 / operand2;
        } else {
          result = Double.NaN; // Indicate division by zero
        }
        break;
      default:
        result = Double.NaN; // Invalid operator
    }
    return result;
  }
}
```

Calculator.jsp

```
<%@ page import="gulakhman.lab9.CalculatorBean" %>
<jsp:useBean id="calculator" class="gulakhman.lab9.CalculatorBean" scope="session"/>

<html>
```

```
<head>
  <title>Simple Calculator</title>
</head>
<body>
<h2>Simple Calculator</h2>
<form method="post" action="calculator.jsp">
  Operand 1: <input type="text" name="operand1" /><br/>
  Operator: <input type="text" name="operator" /><br/>
  Operand 2: <input type="text" name="operand2" /><br/>
  <input type="submit" value="Calculate" /><br/>
</form>

<%
  if (request.getMethod().equalsIgnoreCase("POST")) {
    double operand1 = Double.parseDouble(request.getParameter("operand1"));
    String operator = request.getParameter("operator");
    double operand2 = Double.parseDouble(request.getParameter("operand2"));

    calculator.setOperand1(operand1);
    calculator.setOperator(operator);
    calculator.setOperand2(operand2);
%>
<h3>Result: <%= calculator.getResult() %></h3>
<%
  }
%>
</body>
</html>
```


localhost:8080/lab9_war_exploded/calculator.jsp
Simple Calculator
Operand 1:
Operator:
Operand 2:
Calculate
Result: 0.2


localhost:8080/lab9_war_exploded/calculator.jsp
Simple Calculator
Operand 1: 10
Operator: /
Operand 2: 50
Calculate
Result: 255.0