Lab Assignment-06

ROLL: 2005535 | NAME: SAHIL SINGH | DATE: 17/02/22

QUES 1 : WAP in Java to implement Binary search in a 1D array.

SOLUTION:

class BinarySearch {

*int* binarySearch(*int* *arr*[], *int* *x*) {

*int* l = 0, r = *arr*.length - 1;

        while (l <= r) {

*int* m = l + (r - l) / 2;

            if (*arr*[m] == *x*)

                return m;

            if (*arr*[m] < *x*)

                l = m + 1;

            else

                r = m - 1;

        }

        return -1;

    }

    public static *void* main(String *args*[]) {

        BinarySearch ob = new BinarySearch();

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

        //int n = arr.length;

*int* x = 4;

*int* result = ob.binarySearch(arr, x);

        if (result == -1)

            System.out.println("Element not found!!!");

        else

            System.out.println("Element found at " + "index " + result);

    }

}

OUTPUT:

Element found at index 3

QUES 2: Illustrate the execution of constructors in multi-level inheritance with three Java classes – plate(length, width), box(length, width, height), wood box(length, width, height, thickness).

SOLUTION:

import java.util.Scanner;

class plate {

*int* length, width;

    plate() {

        System.out.println("Parent Class Constructor Called!!!");

    }

}

class box extends plate {

*int* height;

    box() {

        System.out.println("Child Class Constructor Called!!!");

    }

}

class wood\_box extends box {

*int* thickness;

    wood\_box() {

        System.out.println("Grandchild Class Constructor Called!!!");

    }

}

class MLI {

    public static *void* main(String[] *args*) {

        Scanner sc = new Scanner(System.in);

        wood\_box obj = new wood\_box();

        sc.close();

    }

}

OUTPUT:

Parent Class Constructor Called!!!

*Child* *Class* *Constructor* Called!!!

*Grandchild* *Class* *Constructor* Called!!!

QUES 3: Java Program to display the lower triangular matrix.

SOLUTION:

public class LowerTriangular {

    public static *void* main(String[] *args*) {

*int* rows, cols;

*int* a[][] = {

                { 1, 2, 3 },

                { 4, 5, 6 },

                { 7, 8, 9 }

        };

        rows = a.length;

        cols = a[0].length;

        if (rows != cols) {

            System.out.println("It should be a square matrix!!!");

        } else {

            System.out.println("Lower triangular matrix: ");

            for (*int* i = 0; i < rows; i++) {

                for (*int* j = 0; j < cols; j++) {

                    if (j > i)

                        System.out.print("  ");

                    else

                        System.out.print(a[i][j] + " ");

                }

                System.out.println();

            }

        }

    }

}

OUTPUT:

Lower triangular matrix:

1

4 5

7 8 9

QUES 4: Java Program to find the transpose of a given matrix.

SOLUTION:

public class MatrixTranspose {

    public static *void* main(String[] *args*) {

*int* original[][] = {

                { 1, 2, 3 },

                { 4, 5, 6 },

                { 7, 8, 9 }

        };

*int* transpose[][] = new *int*[3][3];

        for (*int* i = 0; i < 3; i++) {

            for (*int* j = 0; j < 3; j++) {

                transpose[i][j] = original[j][i];

            }

        }

        System.out.println("Matrix Before Transpose: ");

        for (*int* i = 0; i < 3; i++) {

            for (*int* j = 0; j < 3; j++) {

                System.out.print(original[i][j] + " ");

            }

            System.out.println();

        }

        System.out.println("Matrix After Transpose: ");

        for (*int* i = 0; i < 3; i++) {

            for (*int* j = 0; j < 3; j++) {

                System.out.print(transpose[i][j] + " ");

            }

            System.out.println();

        }

    }

}

OUTPUT:

Matrix Before Transpose:

1 2 3

4 5 6

7 8 9

Matrix After Transpose:

1 4 7

2 5 8

3 6 9

**-----------------------------------------------------------------------------------------**