**PRACTICAL 1.1**

* Introduction to Object Oriented Concepts, comparison of Java with other object oriented programming languages. Introduction to JDK, JRE, JVM, javadoc, command line argument.

🡪Object means a real word entity such as pen, chair, table etc. Object- Oriented Programming is a methodology or paradigm to design a program using classes and objects. It simplifies the software development and maintenance by providing some concepts:

● Object

● Class

● Inheritance

● Polymorphism

● Abstraction

● Encapsulation

🡪**Object**: Any entity that has state and behaviour is known as an object. For example: chair, pen, table, keyboard, bike etc. It can be physical and logical.

🡪**Class**: Collection of objects is called class. It is a logical entity.

🡪**Inheritance**: When one object acquires all the properties and behaviours of parent object i.e. known as inheritance. It provides code reusability. It is used to achieve runtime polymorphism.

🡪**Polymorphism**: When one task is performed by different ways i.e. known as polymorphism. For example: to convince the customer differently, to draw something e.g. shape or rectangle etc. In java, we use method overloading and method overriding to achieve polymorphism. Another example can be to speak something e.g. cat speaks meaw, dog barks woof etc.

🡪**Abstraction**: Hiding internal details and showing functionality is known as abstraction. For example:

phone call, we don't know the internal processing.

In java, we use abstract class and interface to achieve abstraction.

🡪**Encapsulation**: Binding (or wrapping) code and data together into a single unit is known as encapsulation. For example: capsule, it is wrapped with different medicines.

A java class is the example of encapsulation. Java bean is the fully encapsulated class because all the data members are private here.

🡪**Difference between JDK, JRE and JVM**

Understanding the difference between JDK, JRE and JVM is important in Java. We are having brief overview of JVM here.

If you want to get the detailed knowledge of Java Virtual Machine, move to the next page. Firstly, let's see the basic differences between the JDK, JRE and JVM.

🡪**JVM**

JVM (Java Virtual Machine) is an abstract machine. It is a specification that provides runtime environment in which java bytecode can be executed. JVMs are available for many hardware and software platforms. JVM, JRE and JDK are platform dependent because configuration of each OS differs. But, Java is platform independent. The JVM performs following main tasks:

● Loads code

● Verifies code

● Executes code

● Provides runtime environment

**🡪JRE**

JRE is an acronym for Java Runtime Environment. It is used to provide runtime environment. It is the implementation of JVM. It physically exists. It contains set of libraries + other files that JVM uses at runtime.

Implementations of JVMs are also actively released by other companies besides Sun Micro Systems

🡪**JDK**

JDK is an acronym for Java Development Kit. It physically exists. It contains JRE + development tools.

**PRACTICAL 1.2**

**🡪Class file**

import java.util.\*;

public class Darsh1\_2 {

    Darsh1\_2(String a){//making an constructor of class file.

        if(a.charAt(0)=='O' || a.charAt(0)=='o')

        {

            System.out.print(a.charAt(0));//Checking that 'o' is at first bit or not.

        }

        if(a.charAt(1)=='z' || a.charAt(1)=='Z')//Checking that 'z' is at 2nd bit or not.

        {

            System.out.print(a.charAt(1));

        }

    }

}

**🡪main file**

// Name :- Aswani Darsh

// Roll-no :-21ce006

// Aim :-Given a string, return a string made of the first 2 chars (if present), however include

// first char only if it is 'o' and include the second only if it is 'z', so "ozymandias"

// yields "oz".

// startOz("ozymandias") → "oz"

// startOz("bzoo") → "z"

// startOz("oxx") → "o"

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

import java.util.\*;

public class Darsh1\_2main {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter String:");

        String A=sc.nextLine();

        Darsh1\_2 d2 = new Darsh1\_2(A);//calls the constructor of the class file

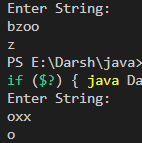
        sc.close();

    }

}

**🡪Output**



****

**PRACTICAL 1.3**

🡪Class file

import java.util.\*;

public class Darsh1\_3

{

    Darsh1\_3(int a,int b)

    {

        int r1,r2;

        r1 = a%10;

        r2 = b%10;//taking remainder of both numbers.

        boolean p=true;

        if(r1 == r2)

        {

            p=true;//checking if the remainders are same the both have same last digit.

        }

        else if(r1!=r2)

        {

            p=false;

        }

        System.out.println("lastDigit("+a+","+b+") --> "+p);

    }

}

🡪main file

// Name :- Aswani Darsh

// Roll-no :-21ce006

// Aim :-Given two non-negative int values, return true if they have the same last digit, such

// as with 27 and 57. Note that the % "mod" operator computes remainders, so 17 %

// 10 is 7.

// lastDigit(7, 17) → true

// lastDigit(6, 17) → false

// lastDigit(3, 113) → true

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

import java.util.\*;

public class Darsh1\_3main {

    public static void main(String args[])

    {

        Scanner s = new Scanner(System.in);

        System.out.print("Enter the first number :");

        int a,b;

        a = s.nextInt();

        System.out.print("\nEnter the second number :");

        b = s.nextInt();

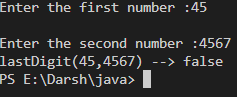
        Darsh1\_3 d3 = new Darsh1\_3(a, b);  //calling the constructor of the class file.

        s.close();

    }

}

🡪output



**PRACTICAL 1.4**

**🡪Class file**

import java.util.Arrays;

public class Darsh1\_4 {

    Darsh1\_4(int a[],int n)

    {

        boolean p=false;

        for(int i=0;i<n;i++)

        {

            if(i==n-1 || i==n-2 || (i==n-3 && a[i]!=1))

            {

                p=false;//checkif that as the array is at last 3 places and not got 1 yet the give false value.

                break;

            }

            else

            {

                if(a[i] == 1)//checks if the curent element if 1 or not

                {

                    if(a[i+1] == 2)//if passes for above the checks that currrent element is 2 or not

                    {

                        if(a[i+2] == 3)//if passes for above the checks that currrent element is 3 or not if yes its true the gives value true to p.

                        {

                            p=true;//

                            break;

                        }

                    }

                }

            }

        }

        System.out.println("array("+Arrays.toString(a)+") -->" +p);

    }

}

**🡪main file**

// Name :- Aswani Darsh

// Roll-no :-21ce006

//  Aim :-Given an array of ints, return true if the sequence of numbers 1, 2, 3 appears in the

// array somewhere.

// array123([1, 1, 2, 3, 1]) → true

// array123([1, 1, 2, 4, 1]) → false

// array123([1, 1, 2, 1, 2, 3]) → true

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

import java.util.\*;

public class Darsh1\_4main {

    public static void main(String[] args) {

        Scanner s =new Scanner(System.in);

        System.out.print("Enter the no of elements you want to be in array :");

        int n = s.nextInt();

        int[] arr = new int[n];

        System.out.println("Enter the array :");

        for(int i=0;i<n;i++)

        {

            arr[i] = s.nextInt();

        }

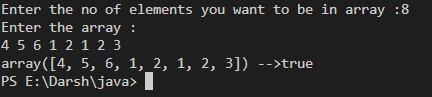
        Darsh1\_4 d4 = new Darsh1\_4(arr,n);//calls constructor of class file

        s.close();

    }

}

**🡪Output**

****

**PRACTICAL 1.5**

🡪Class file

import static java.lang.Math.min;

public class Darsh1\_5 {

    Darsh1\_5(String a, String b) {

        // Figure which string is shorter.

        int len = Math.min(a.length(), b.length());

        int count = 0;

        // Look at both substrings starting at i

        for (int i = 0; i < len - 1; i++) {

            String aSub = a.substring(i, i + 2);// taking substrings of both strings.

            String bSub = b.substring(i, i + 2);

            if (aSub.equals(bSub)) { // Use .equals() with strings

                count++;

            }

        }

        System.out.println("stringMatch(\"" + a + "\", \"" + b + "\") --> " + count);

    }

}

🡪main file

// Name :- Aswani Darsh

// Roll-no :-21ce006

// Aim :-Given 2 strings, a and b, return the number of the positions where they contain the

// same length 2 substring. So "xxcaazz" and "xxbaaz" yields 3, since the "xx", "aa",

// and "az" substrings appear in the same place in both strings.

// stringMatch("xxcaazz", "xxbaaz") → 3

// stringMatch("abc", "abc") → 2

// stringMatch("abc", "axc") → 0

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

import java.util.\*;

public class Darsh1\_5main {

    public static void main(String[] args) {

        Scanner sc=new Scanner(System.in);

        System.out.println("Enter first string : ");

        String s1=sc.next();

        System.out.println("Enter second string : ");

        String s2=sc.next();

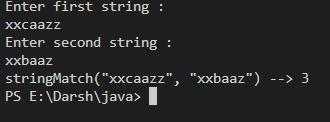
        Darsh1\_5 d5=new Darsh1\_5(s1,s2);

        sc.close();

    }

}

🡪Output



**PRACTICAL 1.6**

🡪class file

import java.util.\*;

public class Darsh1\_6 {

  Darsh1\_6(String[] words, String target) {

    int found = 0;

    for (int i = 0; i < words.length; i++) {

      if (words[i].equals(target)) {

        found++;// checking if any string in the string array is same as the target string.

      }

    }

    found = words.length - found;// removing the length of the string same as target.

    int place = 0;

    String[] str = new String[found];// making a new array for the remaining elements of original array.

    for (int j = 0; j < words.length; j++) {

      if (!words[j].equals(target)) {

        str[place] = words[j];// string element not equal to target are placed in new array.

        place++;

      }

    }

    System.out.println(Arrays.toString(str));

  }

}

🡪main file

// Name :- Aswani Darsh

// Roll-no :-21ce006

// Aim :-Given an array of strings, return a new array without the strings that are equal to

// the target string. One approach is to count the occurrences of the target string, make

// a new array of the correct length, and then copy over the correct strings.

// wordsWithout(["a", "b", "c", "a"], "a") → ["b", "c"]

// wordsWithout(["a", "b", "c", "a"], "b") → ["a", "c", "a"]

// wordsWithout(["a", "b", "c", "a"], "c") → ["a", "b", "a"]

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

import java.util.\*;

public class Darsh1\_6main {

    public static void main(String[] args) {

        Scanner sc = new Scanner(System.in);

        String[] s = new String[5];

        String s1;

        System.out.println("Enter the array of String");

        for(int i=0;i<5;i++)

        {

            s[i] = sc.next();

        }

        System.out.println("Enter the target String");

        s1 = sc.next();

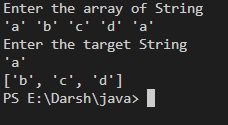
        Darsh1\_6 d6 = new Darsh1\_6(s,s1);

        sc.close();

    }

}

🡪output



**PRACTICAL 1.7**

**🡪class file**

public class Darsh1\_7 {

    private static final String string = " ";

        Darsh1\_7(int nrow){

            int n=0;

            for (int r=0;r<=nrow;r++)

            {

                for (int c=1;c<=nrow-r;c++)

                {

                    System.out.printf("%4s", string);//putting blank spaces until total rows-desired coloumn comes

                }

                for (int c=0;c<=r;c++)

                {

                    n = (int) Math.pow(2, c);//after that prints power of the 2 in the row as per the row.

                    System.out.printf("%4d", n);

                }

                for (int c=r-1;c>=0;c--)

                {

                    n = (int) Math.pow(2, c);

                    System.out.printf("%4d", n);//prints the pow of 2 after the reversed right angle triangle from above is printed to make it a proper pyramid.

                }

                System.out.println();

            }

        }

}

**🡪main file**

// Name :- Aswani Darsh

// Roll-no :-21ce006

// Aim :-Display numbers in a pyramid pattern.

//                 1

//               1 2 1

//             1 2 4 2 1

//           1 2 4 8 4 2 1

//         1 2 4 8 16 8 4 2 1

//       1 2 4 8 16 32 16 8 4 2 1

//   1 2 4 8 16 32 64 32 16 8 4 2 1

// 1 2 4 8 16 32 64 128 64 32 16 8 4 2 1

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

import java.util.\*;

public class Darsh1\_7main {

    public static void main(String[] args)

        {

            int nrow;

            Scanner s = new Scanner(System.in);

            System.out.println("Enter the number one less than the number rows you want in pyramid :");

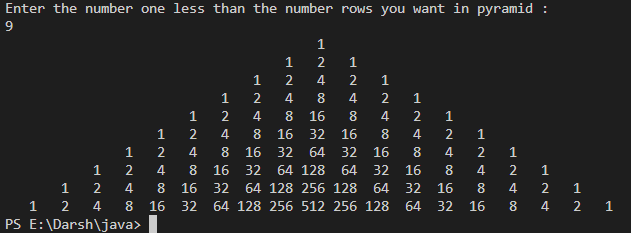
            nrow = s.nextInt();

            Darsh1\_7 d7 = new Darsh1\_7(nrow);//calls the contructor of the class file

        }

}

🡪output



**PRACTICAL 1.8**

**🡪class file**

public class Darsh1\_8 {

    Darsh1\_8(int nstd,int nque,String[] ans,String key)

    {

        int marks[] = new int[nstd];

        char[] c = key.toCharArray();//gives the key to a new string

        for(int i=0;i<nstd;i++)//moves until que of all student be checked

        {

            marks[i]=0;

            char[] c2 = ans[i].toCharArray();//gives the ans given by student to a new string

            for(int j=0;j<nque;j++)

            {

                if(c2[j]==c[j])//compares the key and the ans of the student and increases the marks

                {

                    marks[i]++;

                }

            }

        }

        for(int i=0;i<nstd;i++)

        {

            System.out.println("Marks scored by student "+(i+1)+" :"+marks[i]);

        }

    }

}

🡪main file

// Name :- Aswani Darsh

// Roll-no :-21ce006

// Aim :-The problem is to write a program that will grade multiple-choice tests. Assume

// there are eight students and ten questions, and the answers are stored in a two dimensional array. Each row records a student’s answers to the questions, as

// shown in the following array.

// Students’ Answers to the Questions:

// 0 1 2 3 4 5 6 7 8 9

// Student 0 A B A C C D E E A D

// Student 1 D B A B C A E E A D

// Student 2 E D D A C B E E A D

// Student 3 C B A E D C E E A D

// Student 4 A B D C C D E E A D

// Student 5 B B E C C D E E A D

// Student 6 B B A C C D E E A D

// Student 7 E B E C C D E E A D

// The key is stored in a one-dimensional array:

// Key to the Questions:

// 0 1 2 3 4 5 6 7 8 9

// Key D B D C C D A E A D

// Your program grades the test and displays the result. It compares each student’s

// answers with the key, counts the number of correct answers, and displays it.

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

import java.util.\*;

public class Darsh1\_8main {

    public static void main(String[] args) {

        int nstd,nque;

        Scanner s =new Scanner(System.in);

        System.out.print("Enter the no of Students taken the exams :");

        nstd = s.nextInt();

        System.out.print("Enter the no of question in the exams :");

        nque = s.nextInt();

        String[] ans = new String[nstd];

        System.out.println("Enter the Answers given by the Student :");

        for(int i=0;i<nstd;i++)

        {

            System.out.print("Student "+(i+1)+" :");

            ans[i] = s.next();

        }

        //System.out.println();

        String key;

        System.out.print("Enter the key to the Questions of exam :");

        key = s.next();

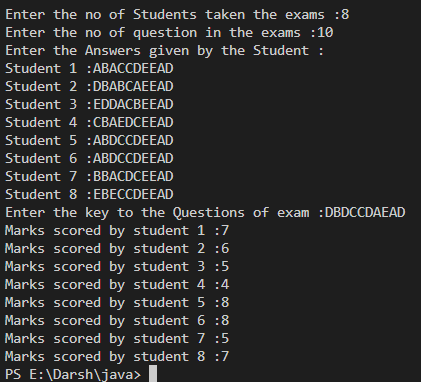
        Darsh1\_8 d8 = new Darsh1\_8(nstd, nque, ans, key);//calls the constructor of the class file.

        s.close();

    }

}

🡪output



**PRACTICAL 1.9**

🡪class file

import java.util.\*;

public class Darsh1\_9 {

    public static boolean notInRow(char arr[][], int row)

    {

        // Set to store characters seen so far.

        HashSet<Character> st = new HashSet<>();

        for(int i = 0; i < 9; i++)

        {

            // If already encountered before,

            // return false

            if (st.contains(arr[row][i]))

                return false;

            // If it is not an empty cell, insert value

            // at the current cell in the set

            if (arr[row][i] != '.')

                st.add(arr[row][i]);

        }

        return true;

    }

// Checks whether there is any duplicate

// in current column or not.

public static boolean notInCol(char arr[][], int col)

{

    HashSet<Character> st = new HashSet<>();

    for(int i = 0; i < 9; i++)

    {

        // If already encountered before,

        // return false

        if (st.contains(arr[i][col]))

            return false;

        // If it is not an empty cell,

        // insert value at the current

        // cell in the set

        if (arr[i][col] != '.')

            st.add(arr[i][col]);

    }

    return true;

}

// Checks whether there is any duplicate

// in current 3x3 box or not.

public static boolean notInBox(char arr[][],int startRow,int startCol)

{

    HashSet<Character> st = new HashSet<>();

    for(int row = 0; row < 3; row++)

    {

        for(int col = 0; col < 3; col++)

        {

            char curr = arr[row + startRow][col + startCol];

            // If already encountered before, return

            // false

            if (st.contains(curr))

                return false;

            // If it is not an empty cell,

            // insert value at current cell in set

            if (curr != '.')

                st.add(curr);

        }

    }

    return true;

}

// Checks whether current row and current column and

// current 3x3 box is valid or not

public static boolean isValid(char arr[][], int row,int col)

{

    return notInRow(arr, row) && notInCol(arr, col) &&

           notInBox(arr, row - row % 3, col - col % 3);

}

public static boolean isValidConfig(char arr[][], int n)

{

    for(int i = 0; i < n; i++)

    {

        for(int j = 0; j < n; j++)

        {

            // If current row or current column or

            // current 3x3 box is not valid, return

            // false

            if (!isValid(arr, i, j))

                return false;

        }

    }

    return true;

}

}

🡪main file

// Name :- Aswani Darsh

// Roll-no :-21ce006

// Aim :-The problem is to check whether a given Sudoku solution is correct.

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

import java.util.\*;

public class Darsh1\_9main {

    public static void main(String[] args) {

        Scanner s = new Scanner(System.in);

        char[][] sudoku = new char[9][9];

        System.out.print("Enter the sudoku you want to be checked :\n");

        for(int i=0;i<9;i++)

        {

            for(int j=0;j<9;j++)

            {

                sudoku[i][j] = s.next().charAt(0);

            }

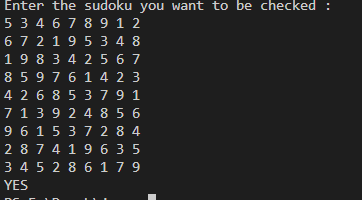
        }

        System.out.println((Darsh1\_9.isValidConfig(sudoku, 9) ? "YES" : "NO"));//calls the method isValidConfig of Darsh1\_9 class file .

    }

}

🡪output



**PRACTICAL 1.10**

🡪class file

public class Darsh1\_10 {

    public static StringBuffer encrypt(String text, int s) {

        StringBuffer result = new StringBuffer();

        for (int i = 0; i < text.length(); i++) {

            if (Character.isUpperCase(text.charAt(i))) //checks if the current char in the entered string is in uppercase or not.

            {

                char ch = (char) (((int) text.charAt(i) + s - 65) % 26 + 65);//if yes then it increases its ascii value by s numbers so the char changes

                result.append(ch);

            }

            else

            {

                char ch = (char) (((int) text.charAt(i) + s - 97) % 26 + 97);//it increases its ascii value by s numbers so the char changes

                result.append(ch);

            }

        }

        return result;

    }

}

🡪main file

// Name :- Aswani Darsh

// Roll-no :-21ce006

// Aim :-Implement ceaser cipher

// Git-hub repository: https://github.com/006Darsh/java-Assaignment-1

public class Darsh1\_10main {

    public static void main(String[] args) {

        String text = "ATTACKATONCE";

        int s = 4;

        System.out.println("Text  : " + text);

        System.out.println("Shift : " + s);

        System.out.println("Cipher: " + Darsh1\_10.encrypt(text, s));//call the encrypt function of Darsh1\_10 class file.

    }

}

🡪output

