

PART-I

Data Types, Variables, Arrays, Operators, Control Statements, String

GitHub Repository Link: <https://github.com/21ce114/JAVA-Practicals.git>

Question 1:	Introduction to Object Oriented Concepts, comparison of Java with other object oriented programming languages. Introduction to JDK, JRE, JVM, javadoc, command line argument
Answer:	<p>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:</p> <ul style="list-style-type: none"> ● Object ● Class ● Inheritance ● Polymorphism ● Abstraction ● Encapsulation <p>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.</p> <p>Class: Collection of objects is called class. It is a logical entity.</p> <p>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.</p> <p>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.</p> <p>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.</p> <p>Abstraction: Hiding internal details and showing functionality is known as abstraction. For example: phone call, we don't know the internal processing.</p> <p>In java, we use abstract class and interface to achieve abstraction.</p> <p>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.</p> <p>A java class is the example of encapsulation. Java bean is the fully encapsulated class because all the data members are private here.</p> <p>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.</p>

	<p>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.</p> <p>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:</p> <ul style="list-style-type: none"> • Loads code • Verifies code • Executes code • Provides runtime environment <p>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.</p> <p>JDK JDK is an acronym for Java Development Kit. It physically exists. It contains JRE + development tools.</p>
Question 2:	<p>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".</p> <p>startOz("ozymandias") → "oz"</p> <p>startOz("bzoo") → "z"</p> <p>startOz("oxx") → "o"</p>
Answer:	<pre> /*ID: 21CE114 Name: Harsh Rana Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git 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".*/ import java.util.*; public class Part1_2 { // Method to check the first letters of the string. public static String checka (String x) { if(x.startsWith("oz")) { return "oz"; } else if(x.startsWith("OZ")) { return "OZ"; } else if(x.startsWith("o")) { return "o"; } } } </pre>

```

    }
    else if(x.startsWith("0")) {
        return "0";
    }
    else if(x.startsWith("z",1)) {
        return "z";
    }
    else if(x.startsWith("Z",1)) {
        return "Z";
    }
    else
        return "";
}

public static void main(String[] args) {

    Scanner sc = new Scanner(System.in);
    //TO take the string from the user.
    String a = sc.nextLine();

    String b = checka(a);

    System.out.println(b);
}

```

Output:

Problems Javadoc Declaration Console ×
 <terminated> Part1_2 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\java.exe
 ozymandias
 OZ

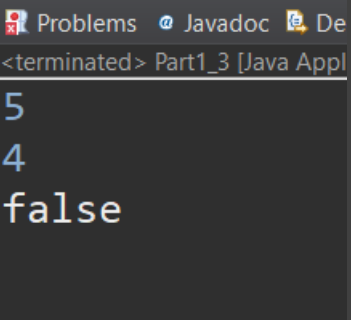
Problems Javadoc Declaration Console ×
 <terminated> Part1_2 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\java.exe
 bzoo
 Z

Problems Javadoc Declaration Console ×
 <terminated> Part1_2 [Java Application] C:\Program Files\Java\jdk-11.0.2\bin\java.exe
 OXX
 O

Question
3:

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

	<code>lastDigit(3, 113) → true</code>
Answer:	<pre>/*ID: 21CE114 Name: Harsh Rana Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git AIM : Given two non-negative <code>int</code> values, return true if they have the same last digit, such as with 27 and 57. Note that the % "<u>mod</u>" operator computes remainders, so 17 % 10 is 7.*/ import java.util.*; public class Part1_3 { public static boolean lastDigit (int x,int y) { //% with 10 will give us the last digit and comparing it with the other last digit number. if((x % 10) == (y % 10)) { return true; } else return false; } public static void main(String[] args) { Scanner <u>sc</u> = new Scanner(System.<u>in</u>); //Taking input for both the numbers from the user. int a=sc.nextInt(); int b=sc.nextInt(); System.out.println(<u>LastDigit</u>(a,b)); } }</pre> <p>Output:</p> 

	
Question 4:	<p>Given an array of ints, return true if the sequence of numbers 1, 2, 3 appears in the array somewhere.</p> <p>array123([1, 1, 2, 3, 1]) → true</p> <p>array123([1, 1, 2, 4, 1]) → false</p> <p>array123([1, 1, 2, 1, 2, 3]) → true</p>
Answer:	<pre> /*ID: 21CE114 Name: Harsh Rana Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git AIM : Given an array of <u>ints</u>, return true if the sequence of numbers 1, 2, 3 appears in the array somewhere.*/ public class Part1_4 { public static boolean array123(int[] x) { //loop to one by one check the whole array. for(int i=0 ; i<x.length-2 ; i++) { //to check weather three consecutive integers are 1,2,3. if (x[i]==1 && x[i+1]==2 && x[i+2]==3) { return true; } } return false; } public static void main(String[] args) { //Test cases provided in the questions. int[] a = {1, 1, 2, 3, 1}; int[] b = {1, 1, 2, 4, 1}; int[] c = {1, 1, 2, 1, 2, 3}; System.out.println(array123(a)); System.out.println(array123(b)); System.out.println(array123(c)); } } </pre> <p>Output:</p>

	 <pre> Problems @ Javadoc Declaration Console × <terminated> Part1_4 [Java Application] C:\Program Files\Java\j true false true </pre>
Question 5:	<p>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.</p> <p>stringMatch("xxcaazz", "xxbaaz") → 3 stringMatch("abc", "abc") → 2 stringMatch("abc", "axc") → 0</p>
Answer:	<pre> /*ID: 21CE114 Name: Harsh Rana Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git 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.*/ public class Part1_5 { public static int stringMatch(String x,String y) { int count=0; //getting the lenght of the smallest string for termination condition in loop. int length=Math.min(x.length(),y.length()); for(int i=0; i<length-1 ;i++) { //storing two letters at the same position in both the string in another sub-strings. String xsub=x.substring(i,i+2); String ysub=y.substring(i,i+2); //comparing the sub strings. if(xsub.equals(ysub)) { count++; } } return count; } public static void main(String[] args) { //test cases from the question. String a="xxcaazz"; String b="xxbaaz"; System.out.println(stringMatch(a,b)); String a1="abc"; String b1="abc"; </pre>

	<pre> System.out.println(stringMatch(a1,b1)); String a2="abc"; String b2="axc"; System.out.println(stringMatch(a2,b2)); } } </pre> <p>Output:</p> 
Question 6:	<p>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.</p> <p>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"]</p>
Answer:	<pre> /*ID: 21CE114 Name: Harsh Rana Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git 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.*/ import java.util.Arrays; public class Part1_6 { public static String[] wordsWithout(String[] x,String y) { //Logic to find the length of the corrected array. int nlen=0; for(int j=0; j<x.length;j++) { if(x[j].equals(y)) { nlen++; } } nlen=x.length-nlen; String[] correct=new String[nlen]; int position=0; //copying the values of the x[] in correct[] except for the target string. for(int i=0 ; i<x.length; i++) { if(!x[i].equals(y)) { correct[position]= x[i]; position++; } } } } </pre>

```

    }
    return correct;
}
public static void main(String[] args) {
    //test Cases given in the question.
    String[] m ={"a", "b", "c", "a"};
    String target = "a";
    //printing all strings in the array using Arrays.toString();
    System.out.println(Arrays.toString(wordsWithout(m, target)));

    String[] m1 ={"a", "b", "c", "a"};
    String target1 = "b";

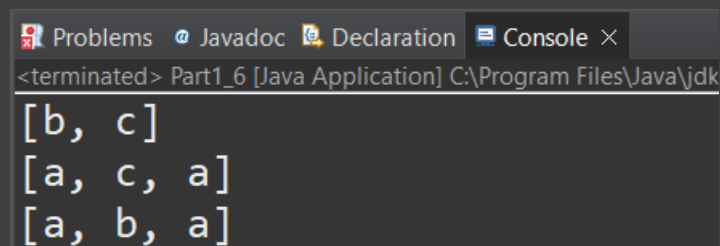
    System.out.println(Arrays.toString(wordsWithout(m1, target1)));

    String[] m2 ={"a", "b", "c", "a"};
    String target2 = "c";

    System.out.println(Arrays.toString(wordsWithout(m2, target2)));
}
}

```

Output:



```

<terminated> Part1_6 [Java Application] C:\Program Files\Java\jdk
[b, c]
[a, c, a]
[a, b, a]

```

Question
7:

Display numbers in a pyramid pattern.

Answer:

```

/*ID: 21CE114
Name: Harsh Rana
Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git
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
*/

```

```

public class Part1_7 {

    public static void main(String[] args) {

        for(int i=1;i<=8;i++)//loop for the total number of rows.
        {

```



```

        for(int j=8;j>i;j--)
        {
            System.out.print(" ");
        }
        int k=1;
        for(int j=1;j<=i;j++)
        {
            System.out.print(k+" ");
            k = k * 2;
        }
        k=k/4;
        for (int j=1;j<i;j++)
        {
            System.out.print(k+" ");
            k = k / 2;
        }
        System.out.println();
    }
}

```

Output:

```

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

```

Question
8:

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

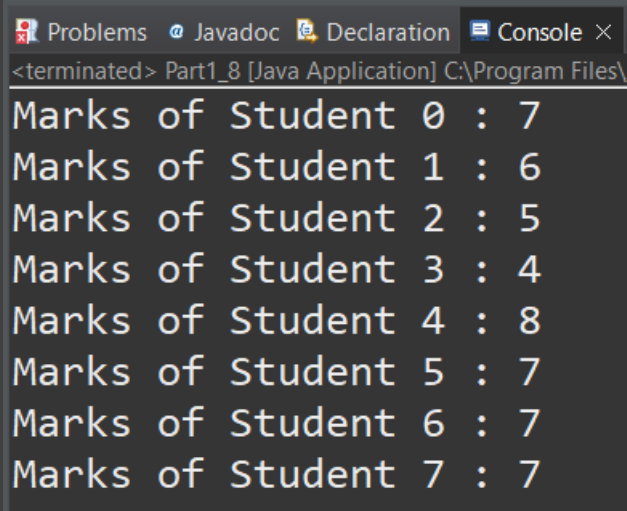
Student 4 A B D C C D E E A D

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:

	<p>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</p> <p>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.</p>
Answer:	<pre> /*ID: 21CE114 Name: Harsh Rana Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git 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.*/ public class Part1_8 { public static void main(String[] args) { //Inputting the marks of different students in 2-D character Array. char[][] marks = { {'A', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'}, {'D', 'B', 'A', 'B', 'C', 'A', 'E', 'E', 'A', 'D'}, {'E', 'D', 'D', 'A', 'C', 'B', 'E', 'E', 'A', 'D'}, {'C', 'B', 'A', 'E', 'D', 'C', 'E', 'E', 'A', 'D'}, {'A', 'B', 'D', 'C', 'C', 'D', 'E', 'E', 'A', 'D'}, {'B', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'}, {'B', 'B', 'A', 'C', 'C', 'D', 'E', 'E', 'A', 'D'}, {'E', 'B', 'E', 'C', 'C', 'D', 'E', 'E', 'A', 'D'}, }; char [] ans = {'D', 'B', 'D', 'C', 'C', 'D', 'A', 'E', 'A', 'D'}; int count = 0; /*The first loop is to access student{row} and the second is for comparing marks{column} with the answer key*/ for(int i = 0 ; i<8; i++) { for(int j = 0; j<10; j++) { if(marks[i][j] == ans[j]) { count++; } } //Printing the count for each student. System.out.println("Marks of Student "+(i)+" : "+count); count = 0; } } } </pre> <p>Output:</p>  <pre> <terminated> Part1_8 [Java Application] C:\Program Files\ Marks of Student 0 : 7 Marks of Student 1 : 6 Marks of Student 2 : 5 Marks of Student 3 : 4 Marks of Student 4 : 8 Marks of Student 5 : 7 Marks of Student 6 : 7 Marks of Student 7 : 7 </pre>

Question 9:

The problem is to check whether a given Sudoku solution is correct.

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

(a) Puzzle

Solution →

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

(b) Solution

Answer:

/*ID: 21CE114

Name: Harsh Rana

Git Repository Link: <https://github.com/21ce114/JAVA-Practicals.git>

AIM : The problem is to check whether a given Sudoku solution is correct.*/

```
public class Part1_9 {
    public static boolean checkSol(int arr[][]){
        boolean b = true; //putting a default boolean value.
        for(int i=0; i<9; i++){
            int sum = 0;
            //Checking if sum of each row is equal to 45.
            for(int j=0; j<9; j++){
                sum = sum + arr[i][j];
            }
            //if the sum is not 45 means number from 1 to 9 are
            repeated somewhere to changing the boolean value and breaking loop.
            if(sum != 45) {
                b = false;
                break;
            }
        }
        return b;
    }
    public static void main(String[] args) {
        //inputting the answer to a sudoku puzzle in 2D array.
        int a[][] = {{ 5, 3, 4, 6, 7, 8, 9, 1, 2 },
            { 6, 7, 2, 1, 9, 5, 3, 4, 8 },
            { 1, 9, 8, 3, 4, 2, 5, 6, 7 },
            { 8, 5, 9, 7, 6, 1, 4, 2, 3 },
            { 4, 2, 6, 8, 5, 3, 7, 9, 1 },
            { 7, 1, 3, 9, 2, 4, 8, 5, 6 },
            { 9, 6, 1, 5, 3, 7, 2, 8, 4 },
            { 2, 8, 7, 4, 1, 9, 6, 3, 5 },
            { 3, 4, 5, 2, 8, 6, 1, 7, 9 }
        };

        System.out.println(checkSol(a));
    }
}
```

Output:

Problems Javadoc Declaration Console ×
 <terminated> Part1_9 [Java Application] C:\Program Files\Java\
 true

Question 10:	Implement Caesar Cipher.
Answer:	<pre> /*ID: 21CE114 Name: Harsh Rana Git Repository Link: https://github.com/21ce114/JAVA-Practicals.git AIM : Implement Caesar Cipher.*/ public class Part1_10 { public static String Encrypt(String text, int shift) { String encryptedText=""; int length = text.length(); //This is because the alphabet only has 26 characters. if(shift > 26) shift = shift % 26; else if(shift < 0) shift = (shift % 26) + 26; for (int i = 0; i < length; i++) { //shifting the characters based on there ASCII value if they are upper case of lower case. char ch = text.charAt(i); if (Character.isLetter(ch)) { if(Character.isUpperCase(ch)) { //If the value end up being out of bound of the ASCII values for the alphabets than going back to the ASCII value for the first alphabet. char c = (char)(ch + shift); if(c > 'Z') encryptedText = encryptedText+(char)(ch - (26 - shift)); else encryptedText = encryptedText+c; } else if (Character.isLowerCase(ch)) { //Same as before just for the lower case character's alphabet's ASCII value. char c = (char)(ch + shift); if(c > 'z') encryptedText += (char)(ch - (26 - shift)); else encryptedText = encryptedText+ c; } } else encryptedText = encryptedText+ch; } return encryptedText; } } </pre>

```

// The whole process is same just doing the reverse steps for the shift value
to decrypt the message.
public static String decrypt(String text, int shift)
{
    String decryptedText="";
    int length = text.length();

    if(shift > 26)
        shift = shift % 26;

    else if(shift < 0)
        shift = (shift % 26) + 26;

    for (int i = 0; i < length; i++)
    {
        char ch = text.charAt(i);
        if (Character.isLetter(ch))
        {
            if(Character.isUpperCase(ch))
            {
                char c = (char)(ch - shift);
                if(c < 'A')
                    decryptedText = decryptedText+(char)(ch +
(26 - shift));

                else
                    decryptedText = decryptedText+c;
            }

            else if (Character.isLowerCase(ch))
            {
                char c = (char)(ch - shift);
                if(c < 'a')
                    decryptedText = decryptedText+(char)(ch +
(26 - shift));

                else
                    decryptedText = decryptedText+ c;
            }

        }

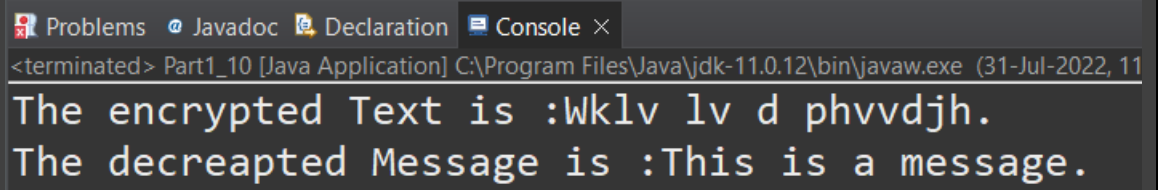
        else
            decryptedText = decryptedText+ch;
    }

    return decryptedText;
}

public static void main(String[] args) {
    //taking the main message that is to be encrypted.
    String text = "This is a message.";
    //taking the value of the shift to be applied to each character.
    //and printing the encrypted text.
    System.out.println("The encrypted Text is :"+Encrypt(text,3));
    //taking the encrypted message to decrypte it.
    String text1 = Encrypt(text,3);
    //taking the value of the shift to be applied to each character.
    //printing the decreapted message.
    System.out.println("The decreapted Message is :"+decrypt(text1, 3));
}
}

```

Output:



The screenshot shows an IDE console window with tabs for Problems, Javadoc, Declaration, and Console. The Console tab is active, displaying the output of a Java application. The output consists of two lines: "The encrypted Text is :Wklv lv d phvvdjh." and "The decreapted Message is :This is a message." The text is in a monospaced font, and the console window has a dark background.

```
<terminated> Part1_10 [Java Application] C:\Program Files\Java\jdk-11.0.12\bin\javaw.exe (31-Jul-2022, 11:00:00 AM)  
The encrypted Text is :Wklv lv d phvvdjh.  
The decreapted Message is :This is a message.
```