ICSE SEMESTER 2 EXAMINATION

SAMPLE PAPER - 5

COMPUTER APPLICATIONS

Maximum Marks: 50

Time allowed: One and a half hours

Answers to this Paper must be written on the paper provided separately.

You will not be allowed to write during the first 10 minutes.

This time is to be spent in reading the question paper.

The time given at the head of this Paper is the time allowed for writing the answers.

---

*Attempt **all** questions from **Section A** and **any four** questions from **Section B.***

SECTION A

*(Attempt **all** questions.)*

Section-A (Attempt all questions)

Question 1.

Choose the correct answers to the questions from the given options. (Do not copy the question, write the correct answer only)

(i) Give the output of the following code :

```
int x=Integer.parseInt("670") + Integer.parseInt ("70");
System.out.println(x)
```

(a) 740 (b) 740.9 (c) 740.0 (d) None of these

(ii) The output of the following statement is :

```
System.out.println(Character.isLetter('a') &&  Character.isUpperCase('t')) ;
```

(a) True (b) False

(iii) The algorithm in which the middle index is found and the array is divided into 2 halves to search for an element is called :

(a) Linear search (c) Long search

(b) Data Search (d) Binary search

(iv) Given a string str=CompuTerScieNceandENGINEERING";

The output of str.substring(5,12) will return _____ characters

(a) 6 (b) 7 (c) 8 (d) 9

(v) If str="ZEBracrossing", The statement System.out.println(str.substring(2,2).toLowerCase()); displays :

(a) b (b) B (c) e (d) No output

(vi) The output of the following code is :

```
String a="45",b="55";
String c = a+b +"10"
int m=Integer.parseInt(a)
int n= Integer.parseInt(b)
```

System.out.println(m+n+Integer.parseInt(c)+"");

(a) 110 (b) 455510100 (c) 455610 (d) None of these

(vii) 1. Given the following values of a and the method doubleLast what will the values of a be after you execute: doubleLast()?

```
private int[ ] a = {-10, -5, 1, 4, 8, 30};
public void doubleLast()
{
    for (int i = a.length / 2; i < a.length; i++)
    {
        a[i] = a[i] * 2;
    }
}
```

(a) {-20, -10, 2, 8, 16, 60} (c) {-10, -5, 1, 8, 16, 60}

(b) {-20, -10, 2, 4, 8, 30} (d) {-10, -5, 1, 4, 8, 30}

(viii) Given a string str= "SoftwareandHardware" The return type of str.length() + str.indexOf('w') will be :

(a) float (b) double (c) int (d) String

(ix) The output of the following code is :

```
public class numbers
{
    public static void main(String[] args)
    {
        int arr[ ]={5,10,15,20,25};
        int s=0;
        for(int i=0;i<arr.length;i++)
        {
            arr[i]=arr[i]-1;
            s=s+arr[i];
        }
        System.out.println(s);
    }
}
```

(a) 70 (b) 69 (c) 50 (d) 80

(x) Given a string str="Barrackpore";

The statement str.substring(5).toUpperCase() returns :

(a) ckpore (b) Barrac (c) CKPORE (d) ackpor

Section-B (Attempt any four questions)

Question 2.

Write a program to accept 10 different decimal numbers (double data type) in a Single Dimensional Array (say, A). Truncate the fractional part of each number of the array A and store their integer part in another array (say, B).

Question 3.

Write a program to input a sentence and display the word of the sentence that contains maximum number of vowels.

Sample Input: HAPPY NEW YEAR

Sample Output: The word with maximum number of vowels: YEAR

Question 4.

Consider the sentence as given below:

Blue bottle is in Blue bag lying on Blue carpet

Write a program to assign the given sentence to a string variable. Replace the word Blue with Red at

all its occurrence. Display the new string as shown below:

Red bottle is in Red bag lying on Red carpet

Question 5.

Write a program in java to initialize two arrays with 5 integers and merge them.

Question 6.

Define a class to input numbers into an array and remove the odd elements.

import java.util.Scanner;

Question 7.

A string is said to be 'Unique' if none of the letters present in the string are repeated. Write a program to accept a string and check whether the string is Unique or not. The program displays a message accordingly.

Sample Input: COMPUTER

Sample Output: Unique String

Section-A

Answer 1.

(i) (a) 740

Explanation :

Integer.parseInt("670") returns 670 and Integer.parseInt ("70") returns 70, 670+70 = 740

(ii) (b) False

Explanation :

Character.isLetter('a') returns "True" , Character.isUpperCase('t') returns "False", True && False gives False

(iii) (d) Binary search

Explanation :

In a binary search the middle index is found and the array is divided into 2 halves to search for an element.

(iv) (b) 7

Explanation :

str.substring(5,12) returns characters from index 5 to 11 which are 7 characters.

(v) (d) No output

Explanation :

str.substring(2,2) returns nothing as start and end index are same.

(vi) (c) 455610

Explanation :

The statement String c = a+b +"10" creates c as concatenation of "45","55" and "10 = 455510. The statement System.out.println(m+n+Integer.parseInt(c)+""); calculates 45 +55 + 455510= 455610

(vii) (c) {-10, -5, 1, 8, 16, 60}

Explanation :

It loops from the middle to the end doubling each value. Since there are 6 elements it will start at index 3.

(viii) (c) int

Explanation :

str.length() returns the length of the string which is an integer and str.indexOf('w') returns the first index of 'w' in the string. Sum of the two is an integer.

(ix) (a) 70

Explanation :

The loop subtracts 1 from each element of the array and sums each of them , which is printed.

(x) (c) CKPORE

Explanation :

str.substring(5) returns "ckpore" and toUpperCase() converts it to CKPORE

Section-B

Answer 2.

```
import java.util.Scanner;
public class TruncateArray
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        double a[] = new double[10];
        int b[] = new int[10];

        System.out.println("Enter 10 decimal numbers");
        for (int i = 0; i < a.length; i++)
        {
            a[i] = in.nextDouble();
            b[i] = (int)a[i];
        }

        System.out.println("Truncated numbers");
        for (int i = 0; i < b.length; i++)
        {
            System.out.print(b[i] + ", ");
        }
    }
}
```

Answer 3.

```
import java.util.Scanner;
public class MaxVowelWord
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.println("Enter a sentence:");
        String str = in.nextLine();
```

```
        str = str + " ";
        String word = "", mWord = "";
        int count = 0, maxCount = 0;
        int len = str.length();

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

            char ch = Character.toUpperCase(str.charAt(i));

            if (ch == 'A' ||
            ch == 'E' ||
            ch == 'I' ||
            ch == 'O' ||
            ch == 'U')
            {
                count++;
            }
            if (ch == ' ')
            {
                if (count > maxCount)
                {
                    maxCount = count;
                    mWord = word;
                }
                    word = "";
                    count = 0;
            }
            else
            {
                word += ch;
            }
        }
        System.out.println("The word with maximum number of vowels: " + mWord);
    }
}
```

Answer 4.

```
public class StringReplace
{
    public static void main(String args[])
    {
        String str = "Blue bottle is in Blue bag lying on Blue carpet";
        str += " ";
        String newStr = "";
        String word = "";
        String target = "Blue";
```

```
            String newWord = "Red";
            int len = str.length();

            for (int i = 0; i < len; i++)
            {
                char ch = str.charAt(i);
                if (ch == ' ')
                {
                    if (target.equals(word))
                    {
                        newStr = newStr + newWord + " ";
                    }
                    else
                    {
                        newStr = newStr + word + " ";
                    }
                    word = "";
                }
                else
                {
                    word += ch;
                }

            }
            System.out.println(newStr);
        }
    }
```

Answer 5.

```
    import java.util.Scanner;
    public class CopyArray
    {
        public static void main(String[] args)
        {

            // array which should be merged
            int src1[] = {10, 20, 30, 40, 50};
            int src2[] = {9, 18, 27, 36, 45};

            // create new array
            int newArray[] = new int[src1.length + src2.length];
            // Copy first to new array from 0 to src1.length
            for(int i=0; i<src1.length; i++)
```

```
        {
            newArray[i] = src1[i];
        }

        // copy second array to new array
        for(int i=0, j=src1.length; j<(src1.length + src2.length); j++, i++)
        {
            newArray[j] = src2[i];
        }

            // display all array

        System.out.println("Array1");
        for (int i-0;i<src1.length;i++)
        System.out.print(src1[i]);
        System.out.print("Array2");
        for (int i-0;i<src2.length;i++)
        System.out.print(src2[i]);
        System.out.print("Merged Array");
        for (int i-0;i< newArray.length;i++)
        System.out.print(newArray[i]);
    }
}
```

Answer 6.

```
public class Array
{
    public static void main(String[] args)
    {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter size of the array: ");
        int n = scan.nextInt();
        int numbers[] = new int[n];
        System.out.println("Enter array elements: ");
        for (int i = 0; i < n; ++i)
        {
                        numbers[i] = scan.nextInt();
        }
            int newArr[] = removeOdd(numbers);
            System.out.println("Array after removing odd numbers: " + Arrays.toString(newArr));
            scan.close();
    }
```

```
    public static int[] removeOdd(int[] numbers)
    {
        int countEven = 0;
        int even[] = null;

        for (int i : numbers)
        {
            if (i % 2 == 0)
            ++countEven;
        }

        even = new int[countEven];

        int i = 0;
        for (int num : numbers)
        {
            if (num % 2 == 0)
            {
                // even
                even[i++] = num;
            }
        }
        return even;
    }
}
```

Answer 7.

```
import java.util.Scanner;
public class UniqueString
{
    public static void main(String args[])
    {
        Scanner in = new Scanner(System.in);
        System.out.print("Enter a string: ");
        String str = in.nextLine();
        str = str.toUpperCase();
        boolean isUnique = true;
        int len = str.length();

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

            char ch = str.charAt(i);
            for (int j = i + 1; j < len; j++)
            {
                if (ch == str.charAt(j))
```

```
                {
                    isUnique = false;
                    break;
                }
            }

            if (!isUnique)
            break;
        }

        if (isUnique)
        System.out.println("Unique String");
        else
        System.out.println("Not Unique String");
    }
}
```

❑❑