ICSE SEMESTER 2 EXAMINATION

SAMPLE PAPER - 2

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) Each primitive data type belongs to a specific:
(a) block (b) object (c) wrapper class (d) none of these

(ii) The process of restricting the free flow of data from the outside world is known as:
(a) Encapsulation (b) Inheritance (c) Function (d) Class

(iii) Given array int x[] = {11, 22, 33, 44}; the value of x[1+2] is _____.
(a) 11 (b) 22 (c) 33 (d) 44

(iv) Which of these is a wrapper for simple data type boolean?
(a) Bool (b) Boolean (c) True (d) False

(v) Which of the following methods checks that a string starts with a particular string or not?
(a) Starts() (b) Startswith() (c) startsWith() (d) StartsWith()

(vi) Given a string str="Publicaccess";
To display the 3rd and 4th characters in uppercase the statement will be :
(a) System.out.println(str.substring(2,4).toUpperCase());
(b) System.out.println(str.substring(2,3).toUpperCase());
(c) System.out.println(str.substring(2).toUpperCase());
(d) System.out.println(str.substring(2,4).touppercase());

(vii) To make accessibility to own class, classes of same package and child classes the access specifier should be :
(a) Private (b) Public (c) Protected (d) Default

(viii) Given a string trn="HimgiriExpress", the statement that will show "EXPRESS" from the string will be :
(a) trn.right(4) (c) trn.substring(7).toUpperCase()
(b) trn.substring(7) (d) trn.right(7,16). toUpperCase()

(ix) The output of the following code is :

```
public class myclass
{
    public static void main(String []args)
    {
        String s1 = new String("Ram");
        String s2 = new String("Laxman");
        System.out.println(s1 = s2);
    }
}
```

(a) Ram (b) Laxman (c) RamLaxman (d) False

Section-B (Attempt any four questions)

Question 2.

Define a class with an integer array to store 6 integers and a main function to accept 6 numbers in the array and swap the adjacent elements.

Question 3.

Given the specification of the following class Student.

Class name : Student

Data members

Roll	integer
Name	String
Term1marks	float
Term2marks	float
Perc	float

Member functions

getStudent() – To input student roll, name and marks in 2 terms and calculate percentage (Formula : Perc=(Term1+Term2)/2)

showStudent() – To display student name and percentage.

Define the class as per the specification.

Question 4.

Define a class to input a sentence and display the words that are palindromes. A palindrome is a word that is same as its reverse.

Question 5.

Define a class to accept 10 strings and a search string. Search for the string and display whether found or not. If not found , display proper message.

Question 6.

Define a class in java to input a sentence and a word and check whether the word is present in the sentence or not.

Question 7.

Defne a class in Java to accept a String/Sentence and display the smallest and longest words present in the String.

Sample Input: "Rahul Dravid is a consistent player"

Sample Output: The longest word: Consistent

The smallest word: is

Answers

Section-A

Answer 1.

(i) (c) Wrapper class

Explanation :

Each primitive data type has a canvas of wrapper class . The wrapper classes have associated functions to deal with the associated data types.

(ii) (a) Encapsulation

Explanation :

The OOPs feature Encapsulation encapsulates data and functions of a class as one unit and shields them from the outside world.

(iii) (d) 44

Explanation :

x[1+2] means x[3] , here x[3] is 44

(iv) (b) Boolean

Explanation :

Boolean is the wrapper class for the Boolean primitive data type.

(v) (c) startsWith()

Explanation :

The startsWith() function checks whether a string begins with a particular string or not and returns a boolean true or false.

(vi) (a) System.out.println(str.substring(2,4).toUpperCase());

Explanation :

The substring(2,4) extracts characters from index 2 to 3 and toUpperCase() converts them to uppercase

(vii) (c) Protected

Explanation :

The protected access specifier can be used to make accessibility to own class, classes of same package and child classes.

(viii) (c) trn.substring(7).toUpperCase()

Explanation :

The trn.substring(7) returns the characters from index 7 to the end of the string i.e : 'Express' and the toUpperCase() function converts it to uppercase to return 'EXPRESS'.

(ix) (b) Laxman

Explanation :

s1=s2 assigns s2 to s1 , which is then printed.

Section-B

Answer 2.

Example :

Input : Enter 6 numbers: 10 20 30 40 50 60

Output : After swap list are: 20 10 40 30 60 50

```
import java.util.Scanner;

class SwapNums
{
    public static void main(String args[])
    {
                int i, t;
                int arr[] = new int[6];
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter 6 numbers:");
                for (i = 0; i < 6; i++)
                {
                    arr[i] = sc.nextInt();
                }
                i = 0;
                while (i < 6 - 1)
                {
                    t = arr[i];
                    arr[i] = arr[i + 1];
                    arr[i + 1] = t;
                    i = i + 2;
                }
                System.out.print("After swap list are:");
                for (i = 0; i < 6; i++)
                {
                    System.out.print(" " + arr[i]);
                }
            }
        }
```

Answer 3.

```
        class  Student
        {
            int  roll;
            String name ;
            float   term1marks;
            float   term2marks;
            float   perc;
            public void  getStudent()
            {
```

```
                Scanner  sc=new Scanner(System.in);
                roll=sc.nextInt();
                name=sc.nextLine();
                term1marks=sc. nextFloat();
                term2marks= sc. nextFloat();
                perc=(term1marks + term2marks)/2;
            }
            public void showStudent( )
            {
                System.out.println("Name  :"+ name)
                System.out.println("Percentage:"+perc);
            }
        }
```

Answer 4.

```
import java.util.Scanner;
public class PalinWords
{
    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 = " ";
        int len = str.length();

        for (int i = 0; i < len; i++)
        {
            char ch = str.charAt(i);
            if (ch == ' ')
            {
                int wordLen = word.length();
                boolean isPalin = true;
                for (int j = 0; j < wordLen / 2; j++)
                {
                    if (word.charAt(j) != word.charAt(wordLen - 1 - j))
                    {
                        isPalin = false;
                        break;
                    }
                }
```

```
                if (isPalin)
                System.out.println(word);

                word = "";
            }
            else
            {
                word += ch;
            }
        }
    }
}
```

Answer 5.

```
import java.util.*;
class search
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        String [] str= new String[10];
        int flag=0,i;
        System.out.println("Enter 10 strings :");
        for(i=0;i<10;i++)
            str[i] = sc.nextLine();
        System.out.println("Enter the name to search :-");
        String s=sc.nextLine();
        for(i=0;i<10;i++)
        {
            if(s==stri])
            {
                flag=1;
                break;
            }
        }
            if(flag==1)
            System.out.println("The name "+s+" Exists");
        else
            System.out.println("The name "+s+" does not Exist");
    }
}
```

Answer 6.

```
import java.util.Scanner;
public class check_words
{
public static void main(String[] args)
{
        String sentence;
        String word;
        String w = "";
        int len, i, f = 0;
        Scanner obj= new Scanner(System.in);
        System.out.println("Enter the sentence: ");  // enter the sentence
        sentence = obj.nextLine();
        sentence = sentence + " ";        // add space at the end of sentence
        len = sentence.length();

    System.out.println("Enter the word to be searched:");  // enter the word
    word = obj.nextLine();
    word = word.trim();
    for(i=0;i<len;i++)
    {
        if(sentence.charAt(i)!=' ')
        {
            w = w + sentence.charAt(i);  // to seperate each word
        }
        else
        {
            if(w.equals(word)){     // check whether the word is the input word
            System.out.println(w+ "is contained in a sentence");
            f=1;
            break;
        }
        w="";      // to reset w to ""
    }
    if(f==0)
    System.out.println(word+ "is not contained in a sentence");
  }
}
```

Answer 7.

```
import java.util.*;
class FindMinMaxString
{
    public static void main(String args[])
    {
        String sen;
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter sentence :");`
        sen=sc.nextLine();
        findMethod(sen);
    }
    static public void findMethod(String s)
    {
        String str = s + " ";
        char ch = ' ';
        int len = str.length(), l = 0;
        int min = len, max = 0;
        String sword = "", lword = "", word = "";
        for (int i = 0; i < len; i++)

                                                {
            ch = str.charAt(i);
            if (ch != ' ')
            {
                word += ch;
            }
                //if ends
            else
            {
                l = word.length();
                if (l < min)

                {
                    min = l;
                    sword = word;
                }
                    //if ends
                    if (l > max)
```

```
                    {
                        max = l;
                        lword = word;
                    }
                    word = "";
                }
            }
            System.out.println("Shortest word = " + sword + " with length " + min);
            System.out.println("Longest word = " + lword + " with length " + max);
        }
}
```

❑❑