ISC SEMESTER 2 EXAMINATION
SAMPLE PAPER - 3
COMPUTER SCIENCE PAPER 1 (THEORY)

Maximum Marks: 35

Time allowed: One and a half hour

*Candidates are allowed an additional **10 minutes** for **only** reading the paper.*

*They must **NOT** start writing during this time.*

---

*Answer **all** questions in **Section A, Section B** and **Section C**.*

While answering questions in Sections A and B, working and reasoning may be indicated briefly.

All working, including rough work, should be done on the same sheet as the rest of the answer.

Section-A

Question 1.

(i) Following is not a part of OOP concept:

(a) encapsulation (b) inheritance (c) recursion (d) abstraction

(ii) Which one of the following is a part of a class?

(a) data members (c) both (a) and (b)

(b) member methods (d) none of them

(iii)
```
void print(int n)
{
    print(n-2);
    if(n<=0)
        return;
    System.out.print(n);
}
```
With reference to the program code given above, what will the function **print( )** returns when the value of n=7 ?

(a) 7531 (b) 7 (c) 531 (d) 1357

(iv) Write the statement in Java to check the word "HOLIDAY" is starting with "H" or not.

(v) What is the logic of Post-order traversal in a Binary Tree?

(vi) What is the output of the statement given below?

System.out.print("EXCELLENT".indexOf('E') + "EXCELLENT".lastIndexOf('E'));

(vii) Give one point of difference between recursion and iteration.

Section-B

Question 2.

Define:

(i) Polymorphism

(ii) Abstract data type

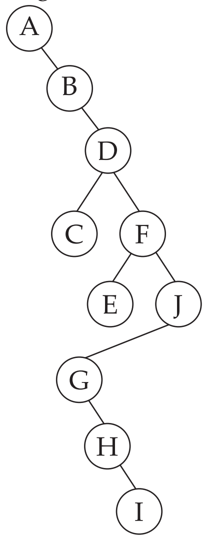
Question 3.

Convert the following infix notation to postfix notation:

(A + B) * (C / D – E / F)

Question 4.

Answer the following questions on the diagram of a Binary Tree given below:



(i) Identify the internal and external nodes of the tree.

(ii) Identify the pairs of siblings.

Section-C

Each program should be written in such a way that it clearly depicts the logic of the problem.
This can be achieved by using mnemonic names and comments in the program.

(Flowcharts and Algorithms are not required.)

The programs must be written in Java.

Question 5.

(i) Design a class **ROUND** which will shift all the letters in a word by one index towards left and the first letter will join at the end.

Example: AFRICA will become FRICAA.

The details of the members of the class are given below:

Class name	:	**ROUND**
Data members/instance variables:		
Wrd	:	stores the input word in uppercase
Res	:	stores the output word
Methods/Member functions:		
ROUND( )	:	default constructor
void acceptword( )	:	to accept the word in uppercase
String round(String s)	:	shift the letters of s as directed above
void display( )	:	displays both the words

Specify the class **ROUND** giving details of the **constructor, void acceptword(), String round(String)** and **void display( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

OR

(ii) Design a class **PigLatin** which will convert the given word to Piglatin form.

A word can be converted to Piglatin form by shifting all the letter from the front to the first occurrence of a vowel to the end followed by "AY"

Example:	TRAIN will become AINTRAY
	MUSIC will become USICMAY

The details of the members of the class are given below:

Class name	:	**PigLatin**
Data members/instance variables:		
Str	:	stores a word in Uppercase
Newstr	:	stores the Piglatin word
Len	:	to store the length of the word
Methods/Member functions:		
PigLatin( )	:	default constructor
void readword( )	:	to accept the word in uppercase
void convert( )	:	convert the input word to its Piglatin form and store it in newstr.
void display( )	:	displays the original word along with the new word

Specify the class **PigLatin** giving details of the **constructor, void readword( ), void convert( )** and **void display( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

Question 6.

(i) A class Series has been defined to generate sum of the following series:

sum = $x + \frac{x^2}{2!} + \frac{x^3}{3!} + \ldots$ upto n terms

Some of the members of the class are given below:

Class name	:	**Series**
Data member/instance variable:		
X	:	integer to store the value of x
N	:	integer to store the value of n (term)
sum	:	double to store the final sum
Member functions/methods:		
Series( )	:	default constructor
void accept( )	:	to accept the value of x and n.
int factorial(int a)	:	return the factorial of a using recursion.
void display( )	:	call the factorial() method and compute the sum of the series.

Specify the class **Series**, giving details of the **Constructor, void accept( ), int factorial(int)**, and **void display( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

OR

(ii) A class **Palindrome** has been defined to check & print the number is a palindrome

A palindrome number is a number whose reverse is the same. For example: 121, 1331, 14041 etc.

Class name	:	**Palindrome**
Data member/instance variable:		
num	:	integer to store the reverse number
rev	:	integer to store the number
Member functions/methods:		
Palindrome( )	:	default constructor
void accept( )	:	to accept a positive number.
int reverse(int a)	:	return the reverse of the given number using recursive technique.
void display( )	:	call the reverse( ) method and check and print the number is palindrome or not and print appropriate message.

Specify the class **Palindrome**, giving details of the **Constructor, void accept( ), int reverse(int)**, and **void display( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

Question 7.

A super class **VEHICLE** stores the details of a vehicle. Another class **CAR** is derived from VEHICLE which performs certain operations.

The details of the **base class** :

Class Name	:	**VEHICLE**
Data member/ instance variable :		
String regnum	:	To store the registration number [alphanumeric].
char Permit	:	The national permit category [either N, S or L].
Member functions/ methods :		
VEHICLE(…)	:	parameterized constructor to initialize the data members.
void Print( )	:	to display the member data.

The details of the **derived class :**

Class Name	:	**CAR**
Data member/ instance variable :		
long Price	:	to store the price of the car.
double Exc	:	to store the excise to be levied on the car.
Member functions/ methods :		
CAR(…)	:	parameterized constructor to initialize the data members of the base and current class. The excise amount (Exc) is initialized to null.
void CalEx()	:	calculates the excise amount as per the given chart.

Permit	Excise Amount
N	22 % of Price
S	11% of Price
L	5% of Price

void Print()	:	to display all the details of both the classes with proper messages (apply method overriding).

Specify the class CAR giving the details of its **mentioned methods**. Assume that class VEHICLE is **already present**.

The super class, main function and algorithm need NOT be written.

Question 8.

A Circular queue is a linear data structure which works on the principle of FIFO, enables the user to enter data from the rear end and remove data from the front end with the rear end connected to the front end to form a circular pattern.

Define a class **CirQueue** with the following details :

Class name		**CirQueue**
Data members/instance variables :		
cq[ ]	:	array to store the integers
cap	:	stores the maximum capacity of the array
front	:	to point the index of the front end
rear	:	to point the index of the rear end

Member functions:

CirQueue(int max)	:	constructor to initialize the data member cap=max, front=0 and rear=0
void push(int n)	:	to add integer in the queue from the rear end if possible, otherwise display the message "QUEUE IS FULL"
int pop( )	:	removes and returns the integer from the front end of the queue if any, else returns -9999
void show( )	:	displays the queue elements

Specify the class **CirQueue** giving details of the functions **void push(int)** and int **pop( )**. Assume that the other functions have been defined.

The main( ) function and algorithm need NOT be written.

Answers

Section-A

Answer 1.

(i) (a) recursion

(ii) (c) both (a) and (b)

(iii) (d) 1357

(iv) "HOLIDAY".startsWith("H");

(v) Post-order traversal follows Left->Right->Root traversal in the tree.

(vi) 6

(vii) Recursion uses if else construct whereas iteration uses looping construct

Section-B

Answer 2.

(i) Polymorphism is the ability of an object to take on many forms.

(ii) Abstract data type is a type (or class) for objects whose behaviour is defined by a set of value and a set of operations.

Answer 3.

(A + B) * (C / D – E / F)

AB+ * (CD/ – EF/)

AB+ * (CD/EF/–)

AB+CD/EF/–*

Answer 4.

(i) Internal nodes: B, D, F, J, G, H

External nodes: C, E, I

(ii) Sibling pairs: (C and F), (E and J)

Section-C

Answer 5.

(i)
```
import java.util.Scanner;
class ROUND
{
    String wrd,res;
    public ROUND( )
    {
        wrd=res=””;
    }

    public void acceptword( )
    {
        Scanner sc=new Scanner(System.in);
        System.out.println(“Enter a word”);
        wrd=sc.next( );
    }

    public String round(String s)
    {
        String t=””;
        int len=s.length( );
        for(int i=1; i<len; i++)
        {
            char c=wrd.charAt(i);
            t+=c;
        }
        t+=s.charAt(0);
        return t;
    }

    public void display( )
    {
        System.out.println(wrd);
        res=round(wrd);
        System.out.println(res);
    }

    public static void main(String ar[ ])
    {
        ROUND ob=new ROUND( );
        ob.acceptword( );
        ob.display( );
    }
}
```

(ii)
```
import java.util.Scanner;
class PigLatin
{
    String str, newstr;
    int len;

    public PigLatin( )
    {
        str=newstr="";
        len=0;
    }

    public void readword( )
    {
        Scanner sc=new Scanner(System.in);
        System.out.println("Enter a word in uppercase");
        str=sc.next( ).toUpperCase( );
        len=str.length( );
    }

    public void convert( )
    {
        int i;
        for(i=0; i<len; i++)
        {
            char c=str.charAt(i);
            if(c=='A'||c=='E'||c=='I'||c=='O'||c=='U')
            break;
        }
        newstr=str.substring(i)+str.substring(0,i)+"AY";
    }

    public void display( )
    {
        System.out.println("Original string:"+str);
        convert( );
        System.out.println("New string:"+newstr);
    }

    public static void main(String ar[ ])
    {
        PigLatin Ob=new PigLatin( );
        Ob.readword( );
        Ob.display( );
    }
}
```

Answer 6.

(i)
```
import java.util.*;
class Series
{
    int x, n;
    double sum;
    public Series ( )
    {
        x=n=0;
        sum=0.0;
    }

    public void accept( )
    {
        Scanner sc=new Scanner(System.in);
        System.out.println(“Enter the value of x and n”);
        x=sc.nextInt( );
        n=sc.nextInt( );
    }

    public int factorial(int a)
    {
        if(a==0 || a==1)
            return 1;
        else
            return a*factorial(a–1);
    }

    public void display ( )
    {
        for(int i=1; i<=n; i++)
        {
            int p=factorial(i);
            sum+=Math.pow(x,i)/factorial(i);
        }
        System.out.print(sum);
    }

    public static void main(String ar[ ])
    {
        Series Ob=new Series( );
        Ob.accept( );
        Ob.display( );
    }
}
```

(ii)
```
import java.util.*;
class Palindrome
{
    int num;
    int rev;

    Palindrome( )
    {
        num=rev=0;
    }

    void accept( )
    {
        Scanner sc= new Scanner(System.in);
        System.out.println(“Enter a number “);
        num=sc.nextInt( );
    }

    int reverse(int x)
    {
        if(x<=0)
            return rev;
        else
        {
            rev=rev*10+x%10;
                return reverse(x/10);
        }
    }

    void display( )
    {
        if(reverse(num)==num)
            System.out.println(“Palindrome no”);
        else
            System.out.println(“Non-palindrome no”);
    }

    static void main( )
    {
        Palindrome ob=new Palindrome( );
        ob.accept( );
        ob.display( );
    }
}
```

Answer 7.

```
class CAR extends VEHICLE
{
    long Price; double Exc;
    CAR(Sring reg1, char permit1, long price1)
    {
        super(reg1, permit1);
        Price = price1;
        Exc = 0;
    }
    void CalEx( )
    {
        if(Permit == ‘N’)
        Exc = 22.0/100 * Price;
        else if(Permit == ‘S’)
        Exc = 11.0/100 * Price;
        else if(Permit == ‘L’)
        Exc = 5.0/100 * Price;
    }
    void Print( )
    {
        super.print( );
        System.out.println(“ Price is ` ” + Price); System.out.print(“ Excise is ` ” + Exc);
    }
}
```

Answer 8.

```
class CirQueue
{
    int cq[ ];
    int cap;
    int front, rear;
    public CirQueue(int n)
    {
        cap=Math.abs(n);
        front=rear=0;
        cq=new int[cap];
    }
    public void push(int n)
    {
        if((rear+1)%cap==front)
        {
            System.out.println(“Queue is full now”);
            return;
        }
        cq[rear]=n;
        rear=(rear+1)%cap;
    }
    public int pop()
    {
        if(front==rear)
        {
            System.out.println("Queue has been emptied");
            return –9999;
        }
        int n=cq[front];
        front=(front+1)%cap;
        return n;
    }
}
```

❑❑