

2018 EXAMINATION



OSWAL
PUBLISHERS

ISC

COMPUTER SCIENCE

MODEL SPECIMEN PAPERS



- ISC Syllabus and Examination paper pattern
- Blue Print to accurately answer questions.
- Unsolved Practice Papers also included.
- Updated with the latest ISC 2018 syllabus.

XII

Indian School Certificate Examination

Model Specimen Papers

COMPUTER SCIENCE

Rupa Pandit
MCA

General Instructions for Students

(Paper-I)

(Three hours)

*(Candidates are allowed additional 15 minutes for **only** reading the paper.
They must **NOT** start writing during this time.)*

*Answer **all** questions in **Part I** (compulsory) and **six** questions from **Part II**, choosing **two** questions from Section-A, **two** from Section-B and **two** from Section-C.*

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

The intended marks for questions or parts of questions are given in brackets [].

(All Rights Reserved)



OSWAL PUBLISHERS

Head office : 1/12, Sahitya Kunj, M.G. Road, AGRA-282 002
Phone : (0562) 2527771-4
Order at : contact@oswalpublishers.com, sales@oswalpublishersl.com
Website : www.oswalpublishers.com
Facebook link : <https://www.facebook.com/oswalpublishers>

CONTENTS

Model Specimen Paper 1	3-17
Model Specimen Paper 2	18-31
Model Specimen Paper 3 (Unsolved)	32-35
Model Specimen Paper 4.....	36-49
Model Specimen Paper 5.....	50-64
Model Specimen Paper 6 (Unsolved)	65-69
Model Specimen Paper 7.....	70-84
Model Specimen Paper 8.....	85-98
Model Specimen Paper 9 (Unsolved)	99-103
Model Specimen Paper 10	104-117
Model Specimen Paper 11	118-130
Model Specimen Paper 12 (Unsolved)	131-135
Model Specimen Paper 13	136-149
Model Specimen Paper 14.....	150-162
Model Specimen Paper 15 (Unsolved)	163-166
Model Specimen Paper 16	167-181
Review	182-182
• Boolean Algebra & Digital Circuits	183-192
• Java Programming	193-228

Edition : 2017

ISBN : 978-93-86299-90-1

ISC Model Specimen Papers are available in the following subjects :

HINDI

CHEMISTRY

ACCOUNTS

ENGLISH-I

MATHEMATICS

COMMERCE

ENGLISH-II

BIOLOGY

BUSINESS STUDIES

PHYSICS

ECONOMICS

BIO-TECHNOLOGY

COMPLETE COURSE IN HINDI : XI & XII

COMPLETE COURSE IN ENGLISH-II : XI & XII

: 272.00

MODEL SPECIMEN PAPER-1

Full Marks : 70

Time : 3 Hours

PART-I (20 Marks)

Answer all questions.

While answering questions in this Part, indicate briefly your working and reasoning, wherever required.

Question 1.

- (a) State the law represented by the following proposition and prove it with the help of a truth table : [1]

$$PVP = P$$

- (b) State the Principle of Duality. [1]
- (c) Find the complement of the following Boolean expression using De Morgan's law : [1]
- $$F(a,b,c) = (b' + c) + a$$
- (d) Draw the logic diagram and truth table for a 2 input XNOR gate. [1]
- (e) If $(\sim P \Rightarrow Q)$ then write its : [1]
- (i) Inverse
- (ii) Converse

Question 2.

- (a) What is an interface ? How is it different from a class ? [2]
- (b) Convert the following infix expression to postfix form : [2]

$$P * Q / R + (S + T)$$

- (c) A matrix [5][10] is stored with each element requiring 8 bytes of storage. If the base address at P[0][0] is 1400, determine the address at P [10][7] when the matrix is stored in Row Major Wise. [2]
- (d) (i) What is the worst case complexity of the following code segment : [2]

```
for (int x = 1; x = a; x++)
{
    statements;
}
for (int y = 1; y <= b; y++)
{
    for (int z = 1; z <= c; z++)
    {
        statement;
    }
}
```

- (ii) How would be complexity change if all the three loops went to N instead of a, b and c?
- (e) Differentiate between a constructor and a method of a class. [2]

Question 3.

The following function **magicfun()** is a part of some class. What will the function **magicfun()** return, when the value of $n=7$ and $n=10$, respectively ? Show the dry run/working : [5]

```
int magicfun (int n)
{
    if (n == 0)
        return 0;
    else
        return magicfun(n/2) * 10 + (n % 2);
}
```

PART-II (50 Marks)

Answer *six* questions in this part, choosing *two* questions from Section A, *two* from Section B and *two* from Section C.

SECTION-A

Answer *any two* questions.

Question 4.

(a) Given the Boolean function:

$$F(A, B, C, D) = \Sigma(2, 3, 4, 5, 6, 7, 8, 10, 11) \quad [4]$$

- (i) Reduce the above expression by using 4-variable Karnaugh map, showing the various groups (*i.e.*, octal, quads and pairs). [4]
- (ii) Draw the logic gate diagram for the reduced expression. Assume that the variables and their complements are available as inputs. [1]

(b) Given the Boolean function :

$$F(P, Q, R, S) = \pi(0, 1, 2, 4, 5, 6, 8, 10)$$

- (i) Reduce the above expression by using 4-variable Karnaugh map, showing the various groups (*i.e.*, octal, quads and pairs). [4]
- (ii) Draw the logic gate diagram for the reduced expression. Assume that the variables and their complements are available as inputs. [1]

Question 5.

(a) A school intends to select candidates for an Inter-School Essay Competition as per the criteria given below : [5]

- The student has participated in an earlier competition and is very creative.

Or

- The student is very creative and has excellent general awareness, but has not participated in any competition earlier.

Or

- The student has excellent general awareness and has won prize in an inter-house competition.

The inputs are :

Inputs	
A	participated in a competition earlier
B	is very creative
C	won prize in an inter-house competition
D	has excellent general awareness

(In all the above cases 1 indicates yes and 0 indicates no).

Output : X [1 indicates yes, 0 indicates no for all cases]

Draw the truth table for the inputs and outputs given above and write the POS expression for X (A, B, C, D).

- (b) State the application of a Half Adder. Draw the truth table and circuit diagram for a Half Adder. [4]
- (c) Convert the following Boolean expression into its canonical POS form : [5]

$$F(A, B, C) = (B + C') \cdot (A' + B)$$

Question 6.

(a) What is a Multiplexer ? How is it different from a decoder ? Draw the circuit diagram for a 8 : 1 Multiplexer. [5]

- (b) Prove the Boolean expression using Boolean laws. Also, mention the law used at each step. [3]

$$F = (x' + z) + [(y' + z) \cdot (x' + y)]' = 1$$

- (c) Define maxterms and minterms. Find the maxterm and minterm when : [2]

$$P = 0, Q = 1, R = 1 \text{ and } S = 0$$

SECTION-B

Answer any two questions.

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 7.

A class **Palin** has been defined to check whether a positive number is a Palindrome number or not. [10]

The number 'N' is palindrome if the original number and its reverse are same. Some of the members of the class are given below :

Class Name	: Palin
Data members /instance variables:	
num	: integer to store the number
revnum	: integer to store the reverse of the number
Methods/Member functions :	
Palin()	: constructor to initialize data members with legal initial values
void accept()	: to accept the number
int reverse (int y)	: reverses the parameterized arguments 'y' and stores it in 'revnum' using recursive technique
void check()	: Checks whether the number is a Palindrome by invoking the function reverse() and display the result with an appropriate message.

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

Question 8.

A class **Adder** has been defined to add any two accepted time. [10]

Example :

Time A – 6 hours 35 minutes

Time B – 7 hours 45 minutes

Their sum is – 14 hours 20 minutes (where 60 minutes = 1 hour)

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

Class Name	: Adder
Data member/instance variable:	
a[]	: integer array to hold two elements (hours and minutes)
Member functions / methods :	
Adder()	: constructor to assign 0 to the array elements
void readtime()	: to enter the elements of the array
void addtime (Adder X, Adder Y)	: adds the time of the two parameterized objects X and Y and stores the sum in the current calling object

void disptime() : displays the array elements with an appropriate message (*i.e.*, hours = and minutes =)

Specify the class **Adder** giving details of the constructor(), void readtime(), void addtime (Adder, Adder) and void disptime(). Define the main() function to create objects and call the functions accordingly to enable the task.

Question 9.

A class **SwapSort** has been defined to perform string related operations on a word input. Some of the members of the class are as follows : [10]

Class Name : **SwapSort**

Data member/instance variable:

wrd : to store a word
 len : integer to store length of the word
 swapwrld : to store the swapped word
 sortwrld : to store the sorted word

Member functions/ methods :

SwapSort() : default constructor to initialize data members with legal initial values
 void readword() : to accept a word in UPPER CASE
 Void swapchar() : to interchange/ swap the first and last characters of the word in 'wrld' and stores the new word in 'swapwrld'
 Void sortword() : Sorts the characters of the original word in alphabetical order and stores it in 'sortwrld'.
 voiddisplay() : displays the original word, swapped word and the sorted word

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

SECTION-C

Answer any two questions.

Each program should be written in such a way that it clearly depicts the logic of the problem stepwise.

This can be achieved by using comments in the program and mnemonic names or pseudo codes for algorithms. The programs must be written in Java and the algorithms must be written in general/standard form, wherever required / specified.

*(Flowcharts are **not** required)*

Question 10.

A super class **Product** has been defined to store the details of a product sold by a wholesaler to a retailer. Define a sub class **Sales** to compute the total amount paid by the retailer with or without fine along with service tax. [5]

Some of the members of both the classes are given below :

Class Name : **Product**

Data member/instance variable :

name : stores the name of the product code
 code : integer to store the product code
 amount : stores the total sale amount of the product (in decimals)

Member functions/methods :

product (String n, int c, double p) : parameterized constructor to assign data members name = n, code= c and amount = p
 void show() : displays the details of the data members

Class name :	Sales
Data member/instance variable :	
day	: stores number of days taken to pay the sale amount
tax	: to store the service tax (in decimals)
totamt	: to store the total amount (in decimals)
Member functions / methods :	
Sales (...)	: parameterized constructor to assign values to data members of both the classes
void compute()	: calculates the services tax @ 12.4% of the actual sale amount calculates the fine @ 2.5% of the actual sale amount only if the amount paid by the retailer to the wholesaler exceeds 30 days. calculates the total amount paid by the retailer as (actual sale amount + service tax + fine)
void show()	: displays the data members of super class and the total amount

Assume that the super class Product has been defined. Using the concept of concept of inheritance, specify the class Sales giving the details of the constructor(...), void compute() and void show().

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

Question 11.

Queue is an entity which can hold a maximum of 100 integers. The queue enables the user to add integers from the rear and remove integers from the front.

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

Class Name	: Queue
Data members/instance variables:	
Que[]	: array to hold the integer elements
size	: stores the size of the array
front	: to point the index of the front
rear	: to point the index of the rear
Member functions :	
Queue (int mm)	: constructor to initialize the data size = mm, front = 0, rear = 0
void addele(int v)	: to add integer from the rear if possible else display the message " Overflow "
int delele()	: returns elements from front if present, otherwise displays the message " Underflow " and returns-9999
void display()	: displays the array elements

Specify the class Queue giving details of ONLY the functions void addele(int) and int delele(). Assume that the other functions have been defined.

The main function and algorithm need NOT be written.

[5]

Question 12.

(a) A linked list is formed from the objects of the class **Node**. The class structure of the Node is given below : [2]

```
class Node
{
    int num;
    Node next;
}
```

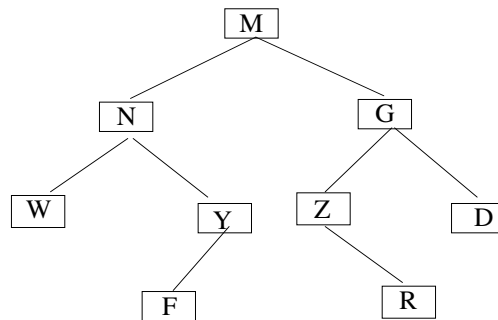
Write an Algorithm **OR** a Method to count the nodes that contain only odd integers from an existing linked list and returns the count.

The method declaration is as follows :

[2]

int CountOdd(Node startPtr)

(b) Answer the following questions from the diagram of a Binary Tree given below :



- (i) Write the postorder traversal of the above tree structure. [1]
- (ii) State the level numbers of the nodes N and R if the root is at 0 (zero) level. [1]
- (iii) List the internal nodes of the right sub-tree. [1]

Answers

PART-I

Answer 1.

(a) It is idempotence law

Truth Table

P	P	PVP
0	0	0
1	1	1
	↑	↑

Verified

(b) According to the principle of Duality, from one Boolean expression, another Boolean expression can be derived by changing :

- (i) All '.' into '+' and '+' into '.'
- (ii) All 0 into 1 and 1 into 0

(c)
$$F(a, b, c) = \overline{(b' + c) + a}$$

$$= \overline{b' + c} \cdot a'$$

$$= (b'' \cdot c') \cdot a'$$

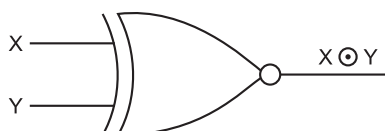
$$= bc' a'$$

$$= a' bc'$$

(d)

X	Y	$X \odot Y$
0	0	1
0	1	0
1	0	0
1	1	1

$$X \odot Y = \bar{X}\bar{Y} + XY$$



(e) ($\sim P \Rightarrow Q$)

- (i) Inverse : $P \Rightarrow \bar{Q}$
- (ii) Converse : $Q \Rightarrow \sim P$

Answer 2.

(a) An interface define a protocol of behaviour.

Interface	Class
(i) It is always implemented.	It is always extended.
(ii) It cannot be instantiated.	It can be instantiated

(b) $PQ^* / R + ST^+$

$PQ^* R / + ST^+$

$PQ^* R / ST^{++}$

(c) $R = 15, C = 10, ES = 8; LR = 0; LC = 0; BA = 1400, i = 10; j = 7$

$P[i][j] = BA + ES (C (i - LR) + (j - LC))$

$P[10][7] = 1400 + 8 (10 (10 - 0) + (7 - 0))$

$= 1400 + 856$

$= 2256$

(d) (i) `for (int x = 1; x <= a; x++)`

`{`
`statement;.....C0`

`}`

`for (int y = 1; y < b; y++)`

`{`

`for (int z = 1; y <= c; z++)`

`{`

`statement;C1`

`}`

`}`

Total time taken :

$C_0 * a + C_1 (b * c)$

$= O(a + bc)$

(ii) $O(N^2)$

Constructor	Method
(i) It does not have a return type, not even void.	It always has a return type.
(ii) It has the same name as that of class name.	It does not have the same name as that of class name.

Answer 3.

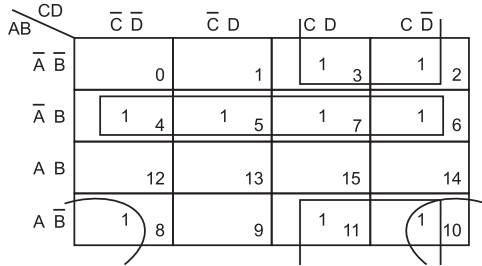
Output	Dry Run		
(i) 111	$n = 7$		
	magicfun (3) * 10 + (1)	} III	
	magicfun (1) * 10 + (1)		II
	magicfun (0) * 10 + (1)		I
	0		

Output	Dry Run		
(ii) 1010	$n = 10$		
	magicfun (5) * 10 + (0)	} IV	
	magicfun (2) * 10 + (1)		III
	magicfun (1) * 10 + (0)		II
	magicfun (0) * 10 + (1)		I

PART-II
SECTION-A

Answer 4.

(a) (i)



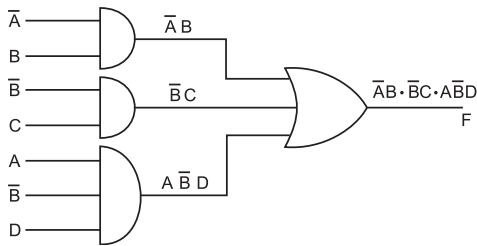
Quad 1 = $m_4 + m_5 + m_6 + m_7 = \bar{A}B$

Quad 2 = $m_2 + m_3 + m_{10} + m_{11} = \bar{B}C$

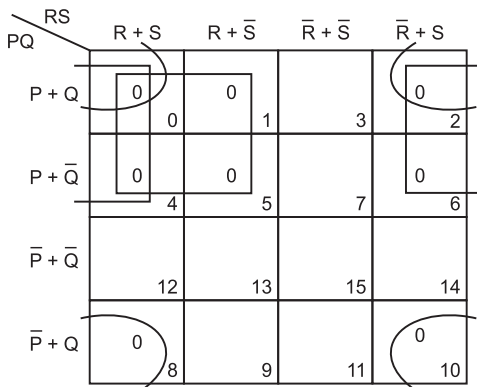
Pair = $m_8 + m_{10} = A\bar{B}\bar{D}$

Reduced Expression – $\bar{A}B + \bar{B}C + A\bar{B}\bar{D}$

(ii)



(b) (i) $F(P, Q, R, S) = \pi(0, 1, 2, 4, 5, 6, 8, 10)$

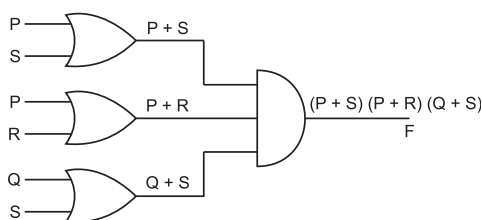


Quad 1 = $M_0 \cdot M_2 \cdot M_4 \cdot M_6 = P + S$

Quad 2 = $M_0 \cdot M_1 \cdot M_4 \cdot M_5 = P + R$

Quad 3 = $M_0 \cdot M_2 \cdot M_8 \cdot M_{10} = (Q + S)$

Reduced Expression = $(P + S)(P + R)(Q + S)$



Answer 5.

(a)

	A	B	C	D	X
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	0
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	1	0	1	1	1
12	1	1	0	0	1
13	1	1	0	1	1
14	1	1	1	0	1
15	1	1	1	1	1

$X(A, B, C, D) = \pi(0, 1, 2, 4, 6, 8, 9, 10)$

$(A + B + C + D)(A + B + C + \bar{D})$

$(A + B + \bar{C} + D)(A + \bar{B} + C + D)$

$(A + \bar{B} + \bar{C} + D)(\bar{A} + B + C + D)$

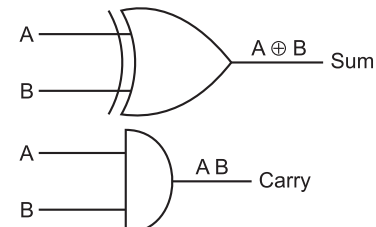
$(\bar{A} + B + C + \bar{D})(\bar{A} + B + \bar{C} + D)$

(b) Half Adder is a Digital electronic circuit which is used to add or manipulate 2 bits.

A	B	Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

Sum : $\bar{A}B + A\bar{B} = A \oplus B$

Carry : AB

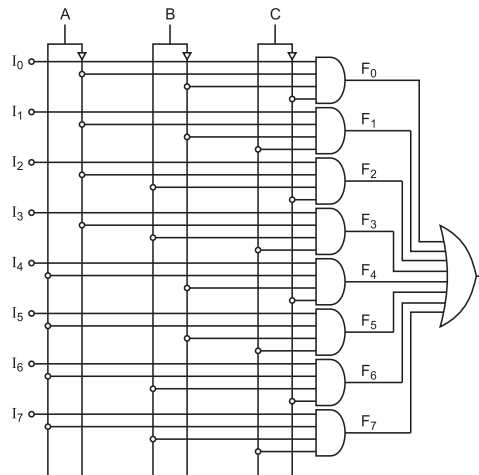


(c) $F(A, B, C) = (B + C')(A' + B)$
 $= (B + C' + 0)(A' + B + 0)$
 $= ((B + C') + AA')(A' + B) + CC'$
 $= (A + B + C')(A' + B + C')(A' + B + C)$
 $= (A + B + C')(A' + B + C')(A' + B + C)$

Answer 6.

- (a) A multiplexer is a digital electronic/ combinational circuit which takes input from one of the many input lines and directs it towards one output line.

Multiplexer	Decoder
It takes input from one of the many input lines and directs it towards one output line by using select lines.	It takes input from n lines and output 2^n or fewer lines.



- (b)
- $$= (x' + z) + \overline{(y' + z)(x' + y)}$$
- [De-Morgan's law]
- $$= (x' + z) + [yz' + xy']$$
- [Distributive (3rd) law]
- $$= x' + xy' + yz' + z$$
- [Complementary law]
- $$= x' + y' + y + z$$
- [Property of 1]
- $$= 1 + x' + z$$
- Hence Proved**

- (c) Maxterms refers to the sum of all the literals with or without bar present within a logic system.

Minterms refer to the product of all the literals with or without bar present within a logic system.

$$P = 0, Q = 1, R = 1, S = 0$$

$$\text{Minterm} = \overline{P}QRS$$

$$\text{Maxterm} = P + \overline{Q} + \overline{R} + S$$

SECTION-B

Answer 7.

```
import java.util.*;
class Palin
{
    int num, revnum;
    public Palin()
    {
        num=0;
        revnum=0;
    }
}
```

```
        void accept()
        {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter a no.");
        num = sc.nextInt();
        }
        int reverse(int y)
        {
        if(y==0)
        {
            return(revnum);
        }
        else
        {
            revnum=revnum * 10 + y%10;
            return( reverse(y / 10));
        }
        }
        void check()
        {
        if(reverse(num)==num)
        {
            System.out.println("Palindrome");
        }
        else
        {
            System.out.println("Not a Palindrome");
        }
        }
        public static void main(String args[])
        {
        Palin ob = new Palin();
        ob.accept();
        ob.check();
        }
    }
```

Answer 8.

```
import java.util.*;
class Adder
{
int a[];
public Adder()
{
    a=new int[2];
a[0]=0;
```

```

a[1]=0;
}
void readtime()
{
    Scanner sc=new
    Scanner(System.in);
    System.out.println("Enter hours and mins");
    a[0]=sc.nextInt();
    a[1]=sc.nextInt();
}
void addtime(Adder X, Adder Y)
{
    a[0]= X.a[0] + Y.a[0];
    a[1]= X.a[1] + Y.a[1];
    if(a[1]>=60)
    {
        a[0]++;
        a[1]-=60;
    }
}
void disptime()
{
    System.out.println("hours = " + a[0] + " and minutes = " + a[1]);
}
public static void main(String args[])
{
    Adder T1=new Adder();
    Adder T2=new Adder();
    Adder T3=new Adder();
    T1.readtime();
    T2.readtime();
    T3.addtime(T1,T2);
    T3.disptime();
}
}

```

Answer 9.

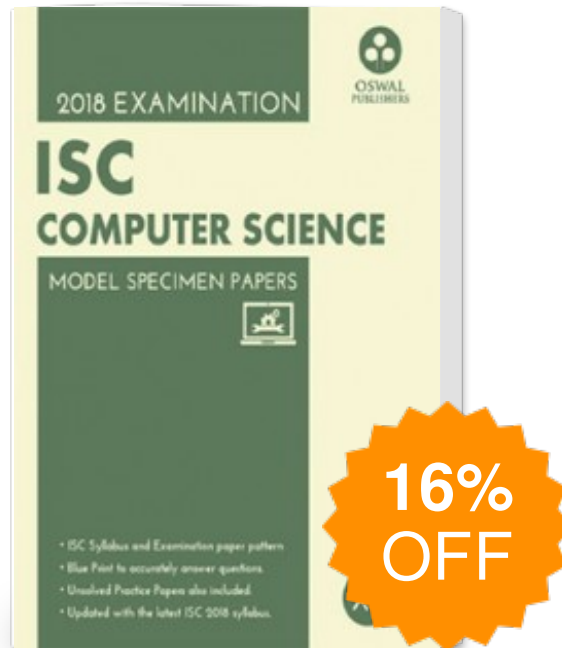
```

import java.util.*;
class SwapSort
{
    String wrd,swapwrd,sortwrd;
    int len;
    public SwapSort()
    {
        wrd=" ";
        len=0;
    }
}

```

```
        swapwrд=""  
        sortwrд=""  
    }  
void readword()  
{  
    Scanner sc=new  
        Scanner(System.in);  
    System.out.println("Enter a word in upper case");  
    wrд=sc.next();  
    len=wrд.length();  
}  
void swapchar()  
{  
    char ch1,ch2;  
    ch1=wrд.charAt(0);  
    ch2=wrд.charAt(len-1);  
    swapwrд=ch2 + wrд.substring(1,len-1) + ch1;  
}  
void sortword()  
{  
    int i,j;  
    for(i=65;i<=90;i++)  
    {  
        for(j=0;j<len;j++)  
        {  
            if(wrд.charAt(j)==i)  
            {  
                sortwrд = sortwrд + wrд.charAt(j);  
            }  
        }  
    }  
}  
void display()  
{  
    System.out.println("Original word " + wrд);  
    System.out.println("Swapped word " + swapwrд);  
    System.out.println("Sorted word " + sortwrд);  
}  
public static void main(String args[])  
{  
    SwapSort ob= new SwapSort();  
    ob.readword();  
    ob.swapchar();
```

Oswal ISC Model Specimen Papers For Class-XII Computer Science 2018 Examination



Publisher : **Oswal Publishers**

ISBN : **9789386299901**

Author : **Panel Of Experts**

Type the URL : <http://www.kopykitab.com/product/13316>



Get this eBook