ISC Paper 2016
Computer Science

Maximum Marks: 70
Time allowed: 3 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-1 (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 [ ].

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 Involution law and prove it with the help of a truth table. [1]

(b) Show that X ∨ ~ (Y ∧ X) is a tautology. [1]

(c) Find the dual of [1]
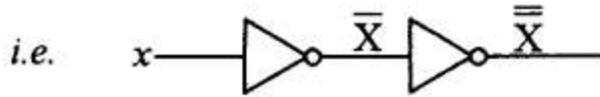YX + X' + 1 = 1

(d) Write the maxterm and minterm, when the inputs are A = 0, B = 1, C = 1 and D = 0. [1]

(e) Draw the logic circuit of a NAND gate using NOR gates only. [1]

Answer:

(a) This law states that $\left(\overline{\overline{X}}\right) = x$

i.e.

Truth Table for $\overline{\overline{x}} = x$

x	$\overline{x}$	$\overline{\overline{x}}$
0	1	0
1	0	1

(b) Show that $X \vee \sim (Y \wedge X)$ is a tautology.

Y	X	Y∧X	~(Y∧X)
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

X	~(Y∧X)	XV ~ (Y∧X)
0	1	1
1	1	1
0	1	1
1	0	1

(c) $Y . X + X' + 1 = 1$

$(Y + X) . (Y') = 1$

(d) $A = 0, \dot{B} = 1, C = 1, D = 0$

Minterm $= \overline{A} \; B \; C \; \overline{D}$

Maxterm $= A + \overline{B} + \overline{C} + D$

(e) A NAND gate can be constructed from the combination of 4 NOR gates.

A NAND
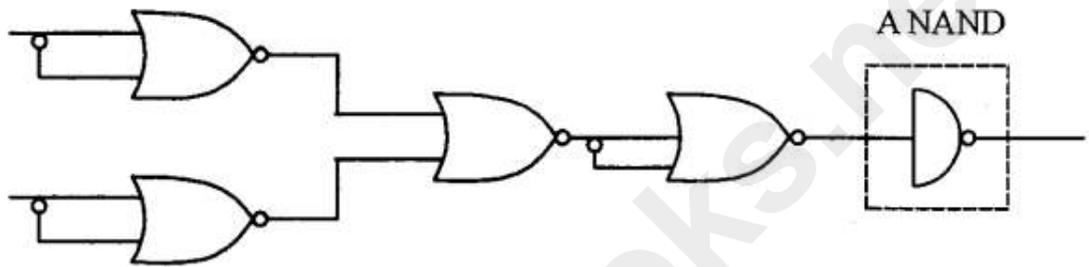
Question 2.
(a) Define the term fall through the condition with reference to switch () case. [2]
(b) Convert the following infix expression into postfix form: [2]
A + B / C * (D/E * F)
(c) A matrix A[m] [n] is stored with each element requiring 4 bytes of storage. If the base address at A[1] [1] is 1500 and the address at A [4] [5] is 1608, determine the number of rows of the matrix when the matrix is stored in Column Major Wise. [2]
(d) From the class declaration given below, state the nature of the identifiers A, B, C and D: [2]
class A extends B implements C, D
(e) State one advantage and one disadvantage of using recursion over iteration. [2]

Answer:
(a) Fall through is prevented with a break keyword at the end of the matching body, which exits execution of the switch block, but this can cause bugs due to unintentional fall through if the programmer forgets to insert break statement.

(b) A + B/C * (D/E * F) = A + B/C (D/EF*)*
= AB/C+ (D/EF*)*
= AB/C + DEF*/*

(c) A [m] [n]
i = 4, B = 1500
j = 5, w = 4
B [i][j] = 1608

(d)
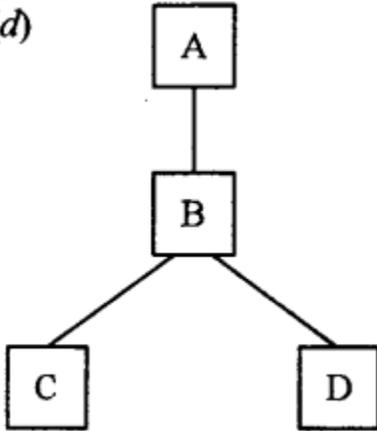
Class A is the superclass of B which in turn is the superclass of subclass C and D

(e) In iteration, the statement is executed repeatedly using the same memory space which is allocated once.
In recursion, the statement is executed repeatedly by invoking the same function within itself and for each recursive call, a fresh memory is allocated. The recursive function runs slower as compared to iteration.

Question 3.
The following function Check() is a part of some class. What will the function Check() return when the values of both 'm' and 'n' is equal to 5? Show the dry run/working. [5]

```
int Check (int m, int n)
{
if(n = = 1)
return - m --;
else
return + + m + Check (m, -- n);
}
```

Answer:
The value of m and n are 5 which is not equal to 1. It is, for this reason, the first if block is not executed as a result it will jump to the else part where the value of m will be 6 and n will be 4. As m is having Pre Increment operator so the value of m is changed to 6 and the value of n to 4 as it is the Pre Decrement operator.

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) = Σ (1, 3, 5, 7, 8, 9, 10, 11, 14, 15).
(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 (A, B, C, D) = π (4, 6, 7, 10, 11, 12, 14, 15)
(i) Reduce the above expression by using the 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]
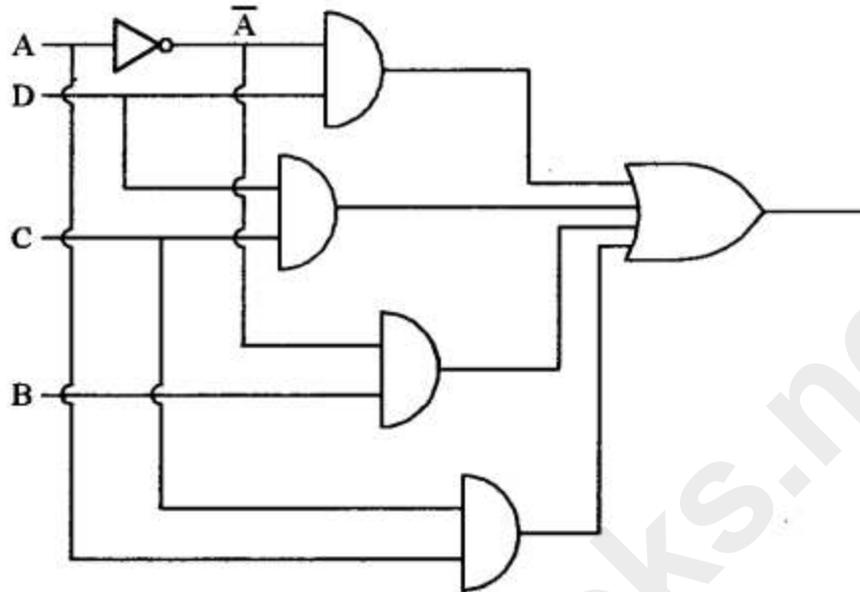
Answer:
(a) (i) F (A, B, C, D) = Σ (1, 3, 5, 7, 8, 9, 10, 11, 14, 15)

From the above expression, we can get four quads.

$$\overline{A}D + CD + \overline{A}B + AC$$

(ii)

(b) (i) $F(A, B, C, D) = \pi(4, 6, 7, 10, 11, 12, 14, 15)$

From the above expression, we can get two quads and one pair.

$BC + AC + B\overline{C}\,\overline{D}$

(ii)
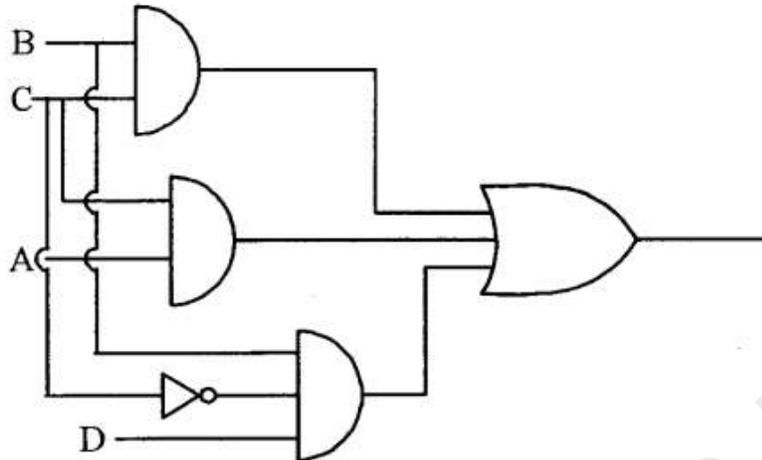
Question 5.
(a) What is a decoder? Draw the logic diagram for a binary to octal (3 to 8) decoder. [3]
(b) How is a half adder different from a full adder? Draw the truth table and derive the SUM and CARRY expression for a full adder. Also, draw the logic diagram for a full adder. [4]
(c) State whether the following expression is a Tautology, Contradiction or a Contingency, with the help of a truth table:
(X=>Z)∨~[(X=>Y)∧(Y=>Z)] [3]

Answer:
(a) A decoder is a circuit which converts the binary number into equivalent decimal form. In a decoder, if there are 3 input lines it will be capable of producing 8 distinct output one for each of the states.

S_2	S_1	S_0	Q_0	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

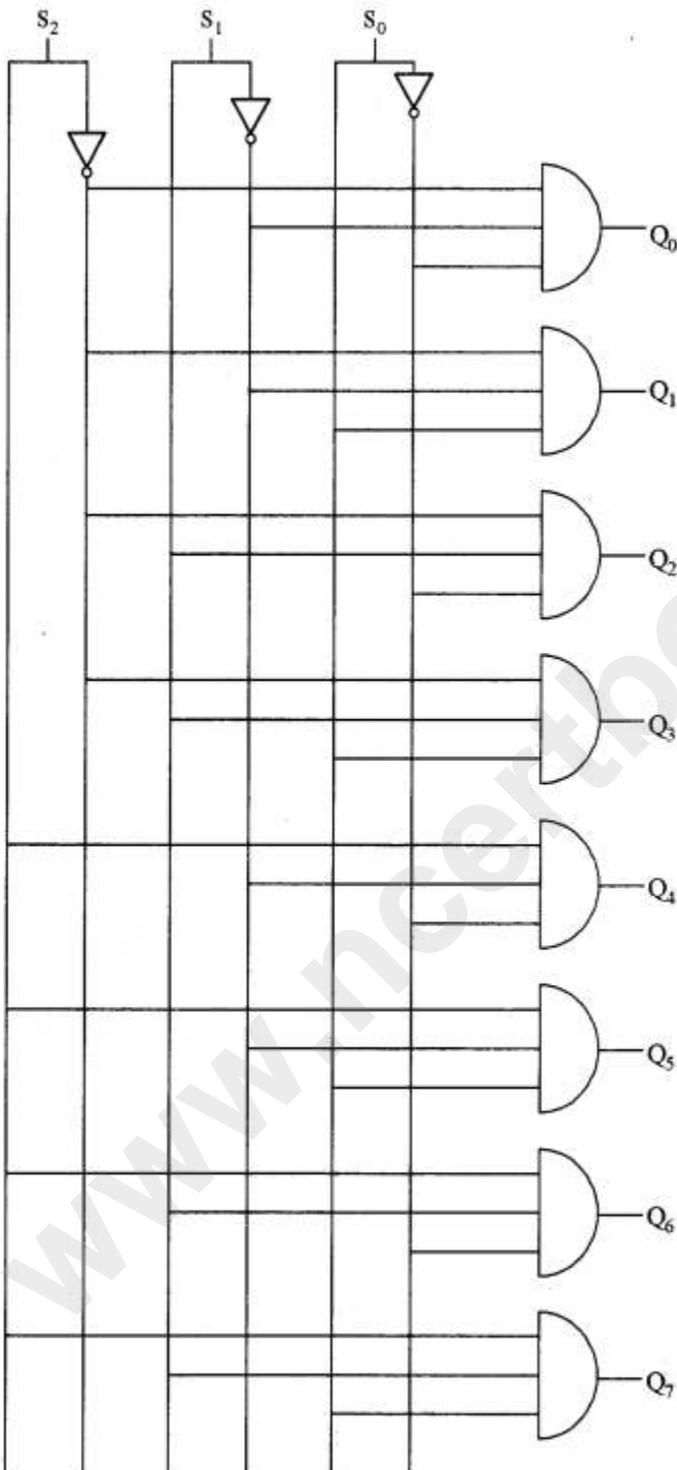Q_0 = 000 Q_2 = 010 Q_4 = 100 Q_6 = 110

Q_1 = 001 Q_3 = 011 Q_5 = 101 Q_7 = 111

Half Adder: It is a logic circuit that adds two bits. It produces the outputs; SUM and CARRY.

Full Adder: It is a logical circuit that adds three bits. It produces two outputs; SUM and CARRY

X	Y	SUM	CARRY
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

$$SUM = X \oplus Y$$

$$CARRY = X.Y$$
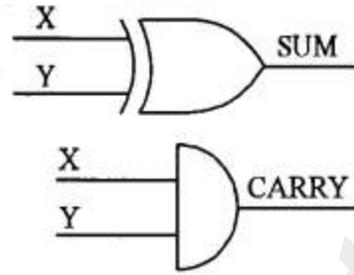
$$SUM = X \oplus Y \oplus C$$

$$CARRY = XZ + YZ + ZX$$
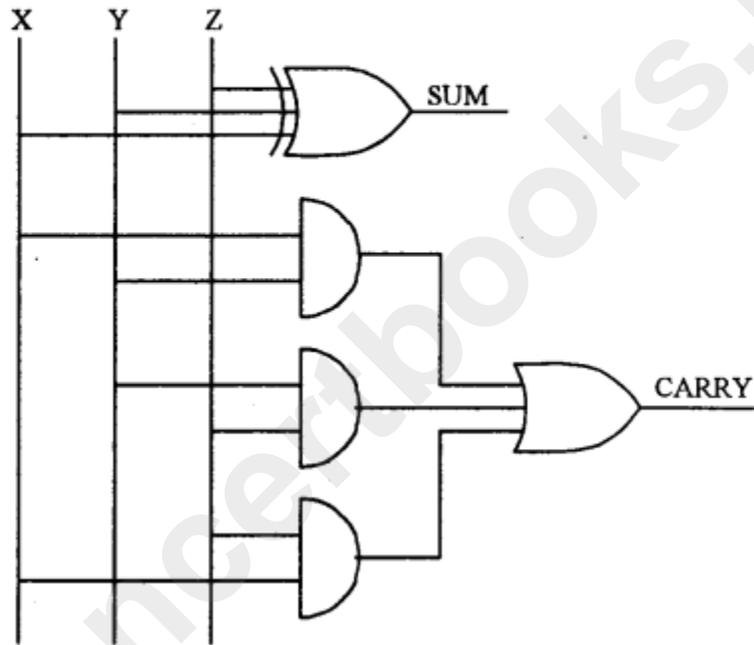
(c) $(X \Rightarrow Z) \lor \sim [(X \Rightarrow Y) \land (Y \Rightarrow Z)]$

X'+Z

X	X'	Z	X'+Z
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1

X'+Y

X	X'	Y	X'+Y
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1

Y'+Z

Y	Y'	Z	Y'+Z
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1

X'+Y	Y'+Z	(X'+Y)∧(Y'+Z)	~
1	1	1	0
1	1	1	0
0	0	0	1
1	1	1	0

X'+Z	~	(X'+Z) V~ [(X'+Y)∧(Y'+Z)]
1	0	1
1	0	1
0	1	1
1	0	1

It is a tautology.

Question 6.
(a) A passenger is allotted a window seat in an aircraft if he/she satisfies the criteria given below: [5]
The passenger is below 15 years and is accompanied by an adult.
or
The passenger is a lady and is not accompanied by an adult.
or
The passenger is not below 15 years but is travelling for the first time
The inputs are:

Inputs	
A	The passenger is below 15 years age.
C	The passenger is accompanied by an adult.
L	The passenger is a lady.

F	The passenger is travelling for the first time.

(In all the above cases 1 indicates yes and 0 indicates no).
Output: W – Denotes the passenger is allotted a window seat (1 indicates yes and 0 indicates no)
Draw the truth table for the inputs and outputs given above and write the SOP expression for W(A, C, L, F).
(b) State the complement properties. Find the complement of the following Boolean expression using De Morgan's law: [3]
AB' + A' + BC
(c) Differentiate between Canonical form and Cardinal form of expression. [2]

Answer:
(a) From the above, we have four inputs and one output W. The truth table for the given variables is as shown below:

A	C	L	F	W
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

The SOP expression of ACLF is :

$$\overline{A}\overline{C}\overline{L}F + \overline{A}\overline{C}LF + \overline{A}C\overline{L}F + \overline{A}C\overline{L}F + \overline{A}CL\overline{F} + \overline{A}CLF + A\overline{C}L\overline{F} + A\overline{C}LF + AC\overline{L}F$$

$$+ AC\overline{L}F + ACL\overline{F} + ACLF$$

(b) The complement property says that any value AND the compliment of that value equals the OR identity and that any value OR the compliment of that value equals the OR identity.
(A AND (A') = 0; A OR (A') = 1.)
Try yourself by applying De Morgan's Law.

(c) Any Boolean function that is expressed as a sum of minterms or as a product of max terms is said to be in its canonical form. The Boolean function which is expressed in the form of degree or cellularity or binary values is referred to as cardinal form.
For example – The following function is in cardinal form.
F(P, Q, R) = π(1, 3)
The cardinal form has been converted into canonical form.
F(P, Q, R) = π(1, 3) = 001, 011
= (P + Q + R'). (P + Q' + R')

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 disarium number is a number in which the sum of the digits to the power of their respective position is equal to the number itself. [10]
Example: 135 = 1^1 + 3^2 + 5^3
Hence, 135 is a disarium number.
Design a class Disarium to check if a given number is a disarium number or not. Some of the members of the class are given below:
Class name: Disarium
Data members/instance variables:
int num: stores the number
int size: stores the size of the number
Methods/Member functions:
Disarium (int nn): parameterized constructor to initialize the data members n = nn and

size = 0

void countDigit(): counts the total number of digits and assigns it to size

int sumofDigits (int n, int p): returns the sum of the digits of the number(n) to the power of their respective positions (p) using recursive technique

void check(): checks whether the number is a disarium number and displays the result with an appropriate message

Specify the class Disarium giving the details of the constructor!), void countDigit(), int sumofDigits(int, int) and void check(). Define the mainO function to create an object and call the functions accordingly to enable the task.

Answer:

```
class Disarium
{
public static void main (String [ ] args) throws IOException
{
BufferedReader br = new BufferedReader (new InputStreamReader) System.in));
System.out.println ("Enter a number");
int n = Integer.parseInt (br.readLine());
int copy = n, d = 0, sum = 0;
String s = Integer.toString (n);
int len = s.length();
while (copy >0)
{
d = copy % 10;
sum = sum + (int) Math.pow (d, len);
len--;
copy = copy/10;
}
if (sum == n)
System, outprintln ("Is a disarium no");
else
System.out.prinln ("Is not a disarium no");
}
}
```

Question 8.
A class Shift contains a two-dimensional integer array of order (m×n) where the maximum values of both m and n are 5. Design the class Shift to shuffle the matrix (i.e. the first row becomes the last, the second row becomes the first and so on). The details of the members of the class are given below: [10]

Class name: Shift

Data member/instance variable:

mat[][]: stores the array element

m: integer to store the number of rows

n: integer to store the number of columns

Member functions/methods:

Shift(int mm, int nn): parameterized constructor to initialize the data members m=mm and n=nn

void input(): enters the elements of the array

void cyclic(Shift p): enables the matrix of the object (P) to shift each row upwards in a cyclic manner and store the resultant matrix in the current object

void display(): displays the matrix elements

Specify the class Shift giving details of the constructor(), void input(), void cyclic(Shift) and void display(). Define the main() function to create an object and call the methods accordingly to enable the task of shifting the array elements.

Answer:

```java
import java.io.*;
class Shift
{
intmat[][];
int m;
int n;
public Shift(int mm, int nn){
m = mm;
n = nn;
m = 5;
n = 5;
mat = new int[m][n];
}
public void input()throws IOException {
InputStreamReader input = new InputStreamReader(System.in);
BufferedReader br = new BufferedReader(input);
for(int i = 0; i < m; i++)
for(int j = 0; j < n; j++){
System.out.print("Elements of the matrix: "+i+", "+j+":");
mat[i] [j] = Integer.parseInt(br.readLine());
}
}
public void cyclic(Shift p){
for(int i = 0; i < m; i++)
for(int j = 0; j < n; j++){
if(i == 0)
this.mat[m-1][j] = p.mat[i][j];
else
this.mat[i-1][j] = p.mat[i][j];
```

```
}
}
public void display() {
for(int i = 0; i < m; i++){
for(int j = 0; j < n; j++)
System, out .print(mat[i] [j]+" \t");
System.out.println();
}
}
public static void main(String args[])throws IOException{
InputStreamReader input = new InputStreamReader(System.in);
BufferedReader br = new BufferedReader(input);
System.out.print("M = ");
int mm=Integer. parseInt(br.readLine());
System.out.print("N = ");
int nn = Integer.parseInt(br.readLine());
Shift obj 1 = new Shift(mm, nn);
Shift obj2 = new Shift(mm, nn);
obj 1.input();
System.out.println("Original Matrix entered by you :");
obj1.display();
obj2.cyclic(obj1);
 System.out.println("New Shifted Matrix is :");
obj2.display();
}
}
```

Question 9.
A class ConsChange has been defined with the following details: [10]
Class name: ConsChange
Data members/instance variables:
word: stores the word len: stores the length of the word
Member functions/methods:
ConsChange(): default constructor
void readword(): accepts the word in lowercase
void shiftcons(): shifts all the consonants of the word at the beginning followed by the vowels (e.g. spoon becomes spnoo)
void changeword(): changes the case of all occurring consonants of the shifted word to uppercase, for e.g. (spnoo becomes SPNoo)
void show(): displays the original word, shifted word and the changed word
Specify the class ConsChange giving the details of the constructor), void readword ( ), void shiftcons (), void changeword() and void show(). Define the main() function to create an object and call the functions accordingly to enable the task.

Answer:

```java
import java.io.*;
class ConsChange {
String word;
int len; public ConsChange() {
word = new String ();
len = word, length ();
}
public void readword () throws IOException {
InputStreamReader isr = new InputStreamReader (System.in);
BufferedReader br = new BufferedReader (isr);
System.out.println ("Enter the word"):
word = br.readLine();
word = word, trim().toLowerCase();
if (word.indexOf (") > 0)
word = word.substring (0, word.indexOf ("));
len = word, length();
}
public void shiftcons()
{
String cons = new String();
String vowel = new String();
for (int i= 0; i < len; i++)
{
char ch = word.charAt (i);
switch (ch)
{
case 'a';
case 'e';
case 'i';
case 'o';
case 'u';
vowel+ = ch;
break;
default:
cons + = ch ;
}
}
word = cons + vowel;
System.out.println ("Shifted Word+word):
changeword();
System.outprintln("Changed Word: "+ word);
}
```

```
public void changeword()
{
int pos = -1;
for (int i = 0; i < len; i++)
char ch = word.charAt (i);
switch (ch)
{
case 'a';
case 'e';
case 'o';
case 'u';
break;
default:
pos = i;
}
word = word.substring (0, pos + 1). toUpperCase ()+word.substring (pos + 1);
}
}
```

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 superclass Bank has been defined to store the details of a customer. Define a subclass Account that enables transactions for the customer with the bank. The details of both the classes are given below: [5]
Class name: Bank
Data members/instance variables:
name: stores the name of the customer
accno: stores the account number
P: stores the principal amount in decimals
Member functions/methods:
Bank(….): parameterized constructor to assign values to the instance variables
void display (): displays the details of the customer
Class name: Account
Data member/instance variable:
amt: stores the transaction amount in decimals

Member functions/methods:

Account(…): parameterized constructor to assign values to the instance variables of both the classes

void deposit(): accepts the amount and updates the principal as p=p+amt

void withdraw(): accepts the amount and updates the principal as p=p-amt

If the withdrawal amount is more than the principal amount, then display the message "INSUFFICIENT BALANCE".

If the principal amount after withdrawal is less than 500, then a penalty is imposed by using the formula.

p=p-(500-p)/10

void display(): displays the details of the customer

Assume that the superclass Bank has been defined.

Using the concept of Inheritance; specify the class Account giving details of the constructor(…), void deposit(), void withdraw() and void display() The superclass and the main function need not be written.

Answer:

```
import java.util.Date
public class Account
{
private int id=0
private double balance = 0
private static double annualRate = 0
private Date dateCreated = new date();
Account ()
{
id = 0;
balance = 0.0;
AnnualInterestRate = 0.0;
}
public Account (int newId, double newBalance)
{
id = newId;
balance = newBalance;
}
Account (int newId, double newBalance, double new AnnualInterestRate)
{
id = newId;
Balance = newBalance;
AnnualInterestRate = new AnnualInterestRate;
}
public int get Id()
{
```

```
return id;
}
public double get Balance()
{
return balance;
}
public double get AnnualInterestRate()
{
return AnnualInterestRate;
}
double get MonthlyInterest Rate()
{
return AnnualInterestRate/12/100 + bal;
}
double withdraw (double amount)
{
return balance - = amount;
}
public double withdraw (double amt, date current date, date preset)
{
if (getBalance() < amount)
{
System.out.println("Amount is larger than current balance");
if (current date.after (preset)
{
System.out.printing("Amount exceed balance");
return super;
withdraw (amount);
}
```

Question 11.
A bookshelf is designed to store the books in a stack with LIFO(Last In First Out)
operation. Define a class Book with the following specifications: [5]
Class name: Book
Data members/instance variables:
name[]: stores the names of the books
point: stores the index of the topmost book
max: stores the maximum capacity of the bookshelf
Methods/Member functions:
Book(int cap): constructor to initialise the data members
max = cap and point = -1
void tell(): displays the name of the book which was last entered in the shelf. If there is
no book left in the shelf, displays the message "SHELF EMPTY"
void add(String v): adds the name of the book to the shelf if possible, otherwise displays

the message 'SHELF FULL"

void display(): displays all the names of the books available on the shelf

Specify the class Book giving the details of ONLY the functions void tell() and void add(String). Assume that the other functions have been defined.

The main function need not be written.

Answer:

```
import java.io.*;
class Book {
String name[];
int point;
int max;
public Book(int cap){
max=cap;
point = -1;
name = new String[max];
}
public void tell(){
if(point == -1)
System.out.println("SHELF EMPTY");
else
System.out.println("Last entered: "+name[point]);
}
public void add(String v){
if (point+1 = max)
System.out.println("SHELF FULL");
else
name [++point] = v;
}
public void display(){
if (point == -1)
System. out.println("SHELF EMPTY");
else {
for(int i = 0; i <= point; i++)
System.out.println(name[i]);
}
}
public static void main(String args[])throws IOException {
InputStreamReader input = new InputStreamReader(System.in);
BufiferedReader br = new BufferedReader(input);
System.out.print("Capacity of the bookshelf: ");
int size = Integer.parseInt(br.readLine());
Book obj = new Book(size);
```

```
loop:
while(true){
System.out.println("1. Last Entered");
System.out.println("2. Add New");
System.out.println("3. List All");
System.out.print("Enter your choice: ");
int choice = Integer.parseInt(br.readLine());
switch(choice){
case 1:
obj.tell();
break;
case 2:
System.out.print("Book title:");
String title=br.readLine();
obj.add(title);
break;
case 3:
obj.display();
break;
default:
break loop;
}
}
}
}
```

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]
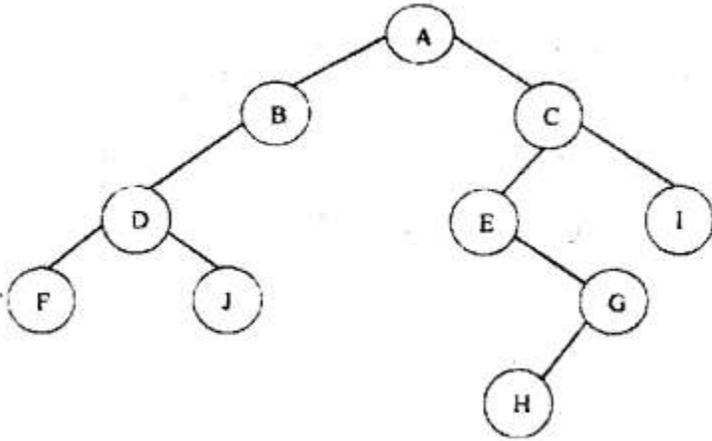
```
class Node
{
String name;
Node next;
}
```
Write an Algorithm OR a Method to search for a given name in the linked list. The method of the declaration is given below:
boolean search name(Node start, String v)
(b) Answer the following questions from the diagram of a Binary Tree given below:

(i) Write the inorder traversal of the above tree structure. [1]
(ii) Name the parent of nodes B and G [1]
(iii) Name the leaves of the right sub-tree. [1]

Answer:

(a) boolean search name(Node start, String v){
if (start == null)
return false;
else if((start.name).equalsIgnoreCase(v))
return true;
return search name(start.next, String v);
}
(b) (i) In the above binary tree, the inorder traversal would be:
F, D, J, B, A, E, H, G, C, I
(ii) The parent of the node B is A, and that of node G is E.
(iii) The leaves of the right sub-tree are H and I.