# STD. XII - CBSE BOARD TEST – 91 (SET - 4)

**Date: 27.03.2017** **COMPUTER SCIENCE - SOLUTIONS**

## SECTION - A

1. (a) new, while, case

(b)
```
#include<isotream.h>
#include<string.h>
```

(c)
```
void main ( )
{
    cout<<"Enter an Alphabate:";
    cin>>CH;
    switch (CH)
{
        case :'A' cout<<"Ant";  Break;
        case :'B' cout<<"Bear: ; Break ;
}
}
```

(d) 22#40#9#13

(e) 10*94*15*74*64*12

(f) (iii) 1 2
2 3

2. (a) **Protected Member:** Protected Members are the members that can be used only by member functions and friends of the class in which it is declared. The protected members are similar to private members that they cannot be accessed by non-member functions. The difference between protected and private becomes clear when we discuss inheritance in chapter 6, because protected members are inheritable but private members are non-inheritable.

**Private Members:** Private Members are the class members that are hidden from the outside world. The private members implement the OOP concept of data hiding. The private members of a class can be used only by member functions (and friends) of the class in which it is declared.

(b) (i) Data hiding and data encapsulation

(ii) Test T1;

Test T2(obj1);

(c)
```
class Box
{
private : int BoxNumber ;
          float side , Area ;
void ExecArea()
{
          Area = Side * Side;
}
public : void GetBox();
         void ShowBox();
};
void Box : : GetBox()
{
      cout <<"enter BoxNumber and side" ;
      cin >> BoxNumber >> side;
}
void Box :: Show Box()
{
      cout << "the BoxNumber and side is" << BoxNumber << Side;
}
void main ()
{
obj . GetBox();
obj.ShowBox();
obj.ExecArea();
}
```

(d) (i) Multilevel Inheritance

(ii) void Display( )

void Enter3 ( )

(iii) T.Display( );

(iv) First Class, Second Class and Third Class will be the order of execution of the constructors, when the object T of class Third is declared inside the main function.

3. (a)

```
void AddUp (int Arr[ ], int N)
{ for (int i = 0 ; i < N ; i ++)
    {
    if (i % 2 = = 0)
        Arr [ i ] = Arr [ i ] + Arr [i + 1];
else
        Arr [ i ] = Arr [ i ] + 10;
}
}
```

(b)

```
void SUMMIDCOL (int MATRIX [5] [3], int N, int M)
{int i, j ;
int sum = 0;
cout << " The array is \ n";
for (i = 0; i < N ; i ++)
{for (j = 0 ; j < M ; j + +)
cout << MATRIX [ i ] [ j ] << "\t";
cout << endl ;
}
for (i = 1 ; i < 2 ; i + +)
{ for (j = 0 ; j < N ; j + +)
{sum = sum + MATRIX [ j ] [ i ] ;
}
cout << \n sum of middle column :" << sum ;
}
}
```

(c) Given:

$Arr[15][20]$

$W = 4 \quad B = ? \quad Lr = 0 \quad Lc = 0 \quad R = 15 \quad C = 20$

Address of $Arr[5][15] = ?$

Address of $[10][5] = 35000$

Address of an element $(I, J)$ in row major $= B + W(C(I - Lr) + (J - Lc))$

Therefore $35000 = B + 4\big(20(10-0)+(5-0)\big)$

$= B + 4(20 \times 10 + 5)$

$= B + 4(200+5)$

$= B + 4 \times 205$

$= B + 820$

$B = 34180$

Address of Arr $[5][15] = 34180 + 4(20 \times 5 + 15)$

$= 34180 + 4(100+15)$

$= 34180 + 4 \times 115$

$= 34180 + 460$

$= 34640$

(d)

```
void STACK : : PUSHGIFT ()
{ GIFT * temp = new GIFT ;
cout << Enter description :" ;
gets(temp → GDESC)
cin > > temp → GCODE ;
temp → next = Top ;
Top = temp ;
}
void STACK : : POPGIFT ()
{if (Top ! = Null)
{GIFT * temp = Top ;
cout << Top → GDESC <<"Deleted";
Top = Top → next ;
delete temp ;
}
else
```

```
cout << " stack is empty" ;
}
```

(e) The expression is : X – (Y + Z) / U * V

$$=\left(X-\left(\left((Y+Z)/U\right)*V\right)\right)$$

The postfix expression is as :

| Operation | Stack Status | Output |
|---|---|---|
| ( | ( | |
| X | ( | X |
| – | ( – | X |
| ( | ( – ( | X |
| ( | ( – ( ( | X |
| ( | ( – ( ( ( | X |
| Y | ( – ( ( ( | XY |
| + | ( – ( ( ( + | XY |
| Z | ( – ( ( ( + | XYZ |
| ) | ( – ( ( | XYZ + |
| / | ( – ( ( / | XYZ + |
| U | ( – ( ( / | XYZ + U |
| ) | ( – ( | XYZ + U / |
| * | ( – ( * | XYZ + U / |
| V | ( – ( * | XYZ + U / V |
| ) | ( – | XYZ + U / V* |
| ) | empty | XYZ + U / V* – |

4. (a)

```
void PURETEXT()
{
ifstream read;
read.open("MYNOTES.TXT");
while(!read.eof())
{
char str=read.get();
for (int i=0;i<str.length();i++)
{
if(str[i])= = 'k')
{
str[i]= 'c';
}
```

```
}
}
```

(b)
```
void COUNTPICS ()
{ PHOTOS P ;
fstream fin ;
fin.open ("PHOTOS.DAT", ios : : binary | ios : : in) ;
char count = 0 ;
while (fin.read ((char*) & P, size of (P)))
{if (P. WhatType ( ) = = 'PORTRAIT')
count ++ ;
}
fin.close ( ) ;
cout << "Total number of photos of type PORTRAIT is " << count ;
}
```

(c) Client Number : 7 of 200

## SECTION - C

5. (a) SELECTION

SELECT * FROM MEMBER WHERE STREAM = "HUMANITIES";

Degree is 3 and Cardinality is 1 of the RESULT.

(b) (i) SELECT * FROM MEMBER ORDER BY ISSUEDATE DESC;

(ii) SELECT DCODE, DTITLE FROM DVD WHERE DTYPE = "Folk";

(iii) SELECT DTYPE, count (ALL DTYPE) FROM DVD;

(iv) SELECT NAME, ISSUEDATE FROM MEMBER WHERE ISSUEDATE >{2016-12-13}

(v)

| MID | NAME | DCODE | ISSUE DATE |
|---|---|---|---|
| 103 | ARTH JOSEPH | F102 | 2016-12-13 |

(vi)

| DTYPE |
|---|
| Rock |
| Folk |
| Classical |

(vii)

| DCODE | NAME | DTITLE |
|---|---|---|
| R102 | AGAM SINGH | A day in life |
| F102 | ARTH JOSEPH | Universal Soldier |
| C101 | NISHA HANS | The Planets |

(viii)

| DTITLE |
|---|
| A day in life |

6. (a) **Demorgan's Law:** This is the most powerful law of Boolean algebra. This states that:

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

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

The truth table for second theorem is:

| X | Y | X.Y | $(X.Y)'$ | X' | Y' | X'+Y' |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 1 | 1 | 1 | 1 |
| 0 | 1 | 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 | 1 | 1 |
| 1 | 1 | 1 | 0 | 0 | 0 | 0 |

To prove algebraically, we know that,

$X + X' = 1$ and $X.X' = 0$

So, if $(X+Y)' = X'.Y'$ then

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

and $(X+Y) + X'.Y' = 0$

Let us prove first part,

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

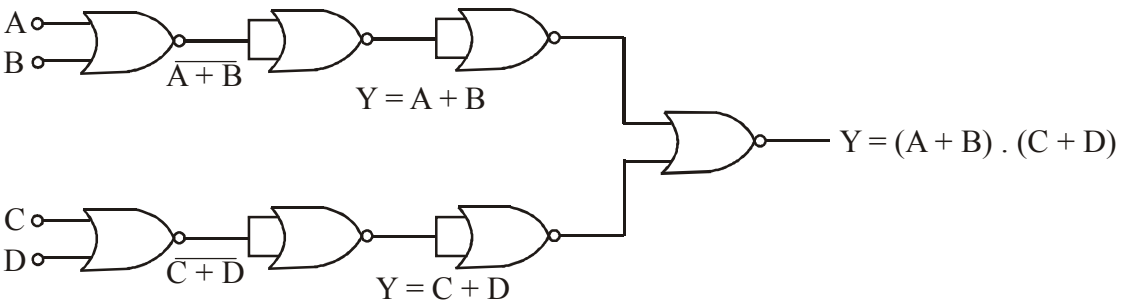
$(X+Y) + X'Y' = ((X+Y)+X').((X+Y)+Y')$ $\quad [(X+Y)(X+Z) = X+Y.Z]$

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

$= (1 + Y)(X + 1)$ $[X + X' = 1 \text{ inverse law}]$

$= 1.1 = 1$ $[1 + X = 1 \text{ identity law}]$

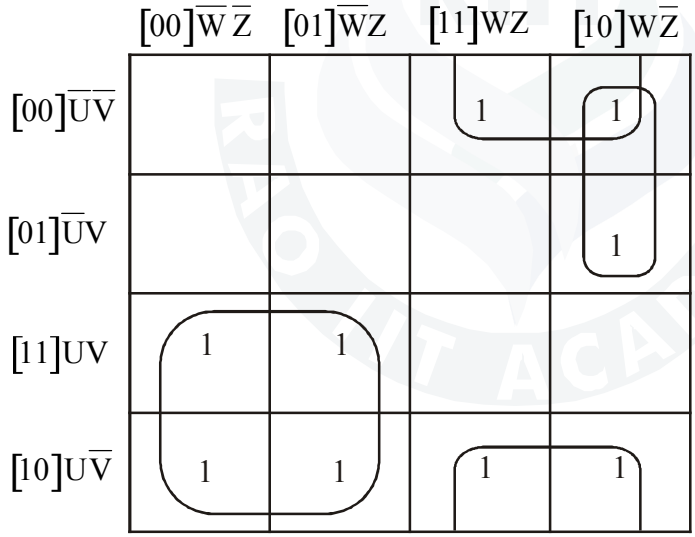
(b)




(c) Canonical POS expression for a Boolean function G

$= (X + Y + Z)(X + Y + \overline{Z})(X + \overline{Y} + \overline{Z})(\overline{X} + \overline{Y} + Z)$

(d)




2 Quads and 1 pair

Quad 1 $(m_2 + m_3 + m_{10} + m_{11})$ reduces to $W\overline{V}$

Quad 3 $(m_8 + m_9 + m_{12} + m_{13})$ reduces to $\overline{W}U$

Pair 1 $(m_2 + m_6)$ reduces to $\overline{U}W\overline{Z}$

$\therefore F(U, V, W, Z) = W\overline{V} + \overline{W}U + \overline{U}W\overline{Z}$

7. (a)

| Characteristic | Description or comparison |
|---|---|
| Speed | Optical fiber transmits data faster than copper Ethernet cable. |
| Effective distance and signal degradation | Optical fiber cable is more effective over longer distances due to the technology used to transmit data. |
| Diameter and weight | Optical fiber stands can be as thin as a human hair and are significantly lighter than their copper Ethernet counterparts. |

(b) (i) Computer Virus:

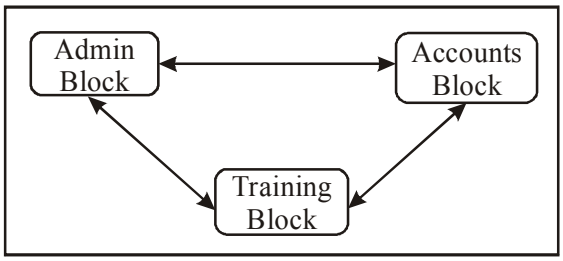
- It can destroy file allocation tables (FAT) and lead to the corruption of an entire file system, resulting in the need to fully reainstall and reload the system.
- It can decrease the space on hard disks by duplicating files.
- It can cause the system to hang so that it does not respond to any keyboard or mouse movements.

(c) (i) Hacker

Hacker is a slang term for a computer enthsiast, i.e., a person who enjoys learning programming languages and computer systems and can often be considered an expert ont he subject(s). The perjorative sense of hacker is becoming more prominent largely because the popular press has co-opted the term to refer to individuals who gain unauthorized access to computer systems for the purpose of stealing and corrupting data. Hackers, themselves, maintain that the proper term for such individuals is cracker. Although hackers still argue that there's a big difference between what they do and what crackers do, the mass media has failed to understand the distinction, so the two terms - hack and crack - are oftern used interchangeably.

(d) (i) Training Block

(ii) The best wired medium is Fibre optic cable.




(iii) Device - Hard Disk Encryptor

Software - Data Masking

(iv) IEEE 802.11 and Wi-Fi router