

CLASS – XII

GUESS PAPER

COMPUTER SCIENCE(Theory) - 083

Time allowed : 3 hours

Maximum marks : 70

Instructions :

- i) **All the questions are compulsory .**
- ii) **Programming Language : C++ .**

1. a) What is the difference between *call by reference* & *call by value* method in a user defined function in C++? Explain it with suitable example. 2

Ans. Call by value : In this method, actual arguments of a calling function gets duplicated as formal arguments & are made available to the called function. As a result whatever change is made by called function in these arguments, are not reflected back in actual arguments.

Ex. void SWAP(float a, float b)

```
{ a=a+b;
b=a-b;
a=a-b;
cout<<"na="<<a<<" and b="<<b; }
```

Call by reference : In this function call, the reference of actual arguments are provided to the called function i.e. memory location of actual arguments. As a result, any change made in these arguments by called function is reflected back to the actual arguments.

Ex. void SWAP(float &a, float &b)

```
{ a=a+b;
b=a-b;
a=a-b; }
```

In the function's argument we simply specify '&' operator along with the argument(s) which are being called by reference.

- b) Write the names of the *header files*, which is/are essentially required to execute the following functions: 1

- i) isdigit()
- ii) sin()

Ans. i) ctype.h **ii)** math.h

- c) Rewrite the following program after removing all the syntactical errors (if any), underlining each correction. : 2

```
include<iostream.h>
typedef char[40] string;
void main( )
{ string S="Australia";
L=strlen(S);
cout<<"String "<<S<<" has '<<L<<"Characters"<<endl; }
```

Ans.

```
#include<iostream.h>
#include<string.h>
typedef char string[40];
```

```
void main( )
{ string S="Australia";
  int/long L=strlen(S);
  cout<<"String "<<S<<" has"<<L<<"Characters"<<endl; }
```

d) Give the **output** of the following program (Assuming that all required header files are included in the program) : 3

```
#define i 5
class TEMP
{ static int a;
  float b;
public:
  TEMP( )
  { b=10; }
  void INTEMP( )
  { a++;
    b=a+10; }
  void OUTTEMP( )
  { cout<<a*i<<"$"<<b-3<<endl; } };
int TEMP::a=2;
void main()
{ TEMP ob[5];
  for(int x=1;x<5;x++)
  ob[x].INTEMP( );
  for(x=1;x<5;x++)
  ob[x].OUTTEMP( );}
```

Ans. 30\$10

30\$11

30\$12

30\$13

e) Give the **output** of the following program (Assuming that all required header files are included in the program) : 2

```
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
void TRANSFER(char *s1,char *s2)
{ int n,j=0;
  for(int i=0;*(s1+i)!='\0';i++)
  {
    n=*(s1+i);
    if(n%2==0)
    *(s2+j++)=*(s1+i);
  } }
void main()
{ char *p="ChaRlesBabBaGe",q[80];
  TRANSFER(p,q);
  cout<<q<<endl;}
```

Ans. hRIBbB

f) Go through the following c++ code, find out the **correct possible output(s)** from the suggested output options i) to iv). Also write the **highest value** which can be assigned to variable G : 2

```
#include<iostream.h>
#include<stdlib.h>
void main()
{
randomize( );
int G,H=5;
G=random(H)+30;
for(int i=35;i>G;i--)
cout<<i<<' '$';
cout<<i;
}
i)    35$34$33$32$31$30$
ii)   35$34$33$32$31
iii)  30$31$32$33$34$35$36
iv)   35$34$33$32$31$30
```

Ans. Options ii) & iv) will be the correct possible outputs. The highest value of variable G would be 34

2. a) What is *constructor overloading*? Support your answer with example. 2

Ans. CONSTRUCTOR OVERLOADING : When more than one constructor appears inside a single class, it is said to be constructor overloading i.e. if we have two or more constructors inside a class, it is said to be an overloaded constructor. For ex.

```
class ABC
{ private:
int x;
float y;
public:
ABC()           //default constructor
{ x=5;
y=0.0; }
ABC(int p, float q) //parameterized constructor
{ x=p;
y=q; }
ABC(ABC &t)      //Copy constructor
{ x=t.p;
y=t.q; }
void INABC( );
void OUTABC( );};
```

Here in the above written example, we see three constructors one after another. Since all of them share the same class name and are different in their type and number of arguments, therefore supports overloading mechanism of OOP.

b) Answer the questions (i) and (ii) after going through the following class : 2

```
class BUS
{ private:
char Pname[30],TicktNo[20];
float Fare;
public:
BUS()           //function 1
{ strcpy(Pname,"\0");
```

```

strcpy(TicktNo, "\0");
Fare=0; }
void Details() //function 2
{ cout<<Pname<<endl<<TicktNo<<endl<<Fare<<endl; }
BUS(char * name, char *tno, float N); //function 3
BUS(BUS &F); // function 4
};

```

- i) In OOP, what is function 3 referred to as? Also define this function.
- ii) Define function 4 and write about its purpose?

Ans. i) Function 3 is referred to as parameterized constructor. Its definition is as follows:

```

BUS(char * name, char *tno, float N)
{ strcpy(Pname,name);
  strcpy(TicktNo,tno);
  Fare=N; }

```

- iii) BUS(BUS &F)


```

      { strcpy(Pname,F.Pname);
        strcpy(TicktNo,F.TicktNo);
        Fare=F.Fare; }
      
```

COPY CONSTRUCTOR : It is used to initialize an instance/object using the values of another instance/object of same class type. It takes the reference to an object of the same class as an argument.

c) Define a class **TAXPAYER** in C++ with following description :

4

Private members :

- a. Name of type string
- b. PanNo of type string
- c. Taxabincm (Taxable income) of type float
- d. TotTax of type double
- e. A function CompTax() to calculate tax according to the following slab:

Taxable Income	Tax%
Up to 160000	0
>160000 and <=300000	5
>300000 and <=500000	10
>500000	15

Public members :

- A parameterized constructor to initialize all the members
- A function INTAX() to enter data for the tax payer and call function CompTax() to assign TotTax.
- A function OUTAX() to allow user to view the content of all the data members.

Ans.

```

class TAXPAYER
{private:
char Name[30],PanNo[30];
float Taxabincm;
double TotTax;
void CompTax()
{ if(Taxabincm >500000)
TotTax= Taxabincm*0.15;
else if(Taxabincm>300000)
TotTax= Taxabincm*0.1;
}
}

```

```

Else if(Taxabincm>160000)
TotTax= Taxabincm*0.05;
else
TotTax=0.0; }
public:
TAXPAYER(char nm[], char pan[], float tax, double ttax) //parameterized constructor
{ strcpy(Name,nm);
strcpy(PanNo,pan);
Taxabincm=tax;
TotTax=ttax; }

void INTAX()
{ gets(Name);
cin>>PanNo>>Taxabincm;
CompTax(); }

void OUTAX()
{ cout<<Name<<'\\n'<<PanNo<<'\\n'<<Taxabincm<<'\\n'<<TotTax<<endl; } };

```

d) Answer the questions (i) to (iv) based on the following :

4

```

class Student
{ private :
    char Rollno[20], Sname[30];
protected :
    auto float marks;
public:
    Student( );
    void ENROL( );
    void SHOW( );
};
class Graduate: public Student
{ char Fname[30];
protected:
    unsigned int age;
public:
    Graduate( );
    void GENROL( );
    void GSHOW( );
};
class Pgraduate: private Graduate
{
    char Mname[25];
    signed int year;
public:
    Pgraduate( );
    void PGENROL( );
    void PGSHOW( );
};

```

i) Mention the member names that are accessible by an object of Pgraduate class.

Ans. PGENROL(), PGSHOW()

ii) Name the data members which can be accessed by the objects of Graduate class.

Ans. None

iii) Name the data members that can be accessed by the functions of Pgraduate class.

Ans. Mname[25], year, age & marks

iv) How many bytes will be occupied by an object of class Pgraduate?

Ans. 113 bytes

3. a) Write a function *TRANSFERP(int ALL[], int N)*, to transfer all the prime numbers from a one dimensional array ALL[] to another one dimensional array PRIME[]. The resultant array PRIME[] must be displayed on screen. **3**

Ans.:

TRANSFERP(int ALL[], int N)

```
{ int PRIME[100],i,j,tp=0,count;
```

```
for(i=0;i<N;i++)
```

```
{
```

```
count=0;
```

```
for(j=0;j<=ALL[i];j++)
```

```
if(ALL[i]%j==0)
```

```
count++;
```

```
if(count==2)
```

```
{
```

```
PRIME[tp]=ALL[i];
```

```
tp++;
```

```
}
```

```
} //end of for
```

//displaying all prime numbers of array PRIME[]

```
cout<<"\nAll prime numbers in resultant array are:\n";
```

```
for(i=0;i<tp;i++)
```

```
cout<<PRIME[i]<<" "; }
```

- b) An array PP[40][32] is stored in the memory along the row with each of the elements occupying 10 bytes. Find out the memory location for the element *PP[18][22]*, if the element *PP[7][10]* is stored at memory location 5000. **3**

Sol.

Given : B=?, W=10, m=40, n=32, I=7, J=10, PP[I][J]=5000, LBr=0, LBc=0

Row Major:

Address of PP[7][10]=B+W(n(I-LBr)+(J-LBc))

$$5000=B+10(32(7-0)+(10-0))$$

$$5000=B+10(224+10)$$

$$5000=B+10*234$$

$$5000=B+2340$$

Therefore $B=5000-2340 = 2660$

Now Address of *PP[18][22] = 2660+10(32(18-0)+(22-0))*

$$=2660+10(576+22)$$

$$=2660+10*598$$

$$=2660+5980 = 8640$$

Ans. Base address = 2660 & address of PP[18][22] is 8640

- c) Write functions to perform *PUSH & POP* operations in a *dynamically allocated stack* containing **4**

the objects of the following structure:

```
struct NODE
{ char name[30];
  float fees;
  NODE *next;  };
```

ANS.:

NODE *top=NULL; //declaring global pointer & initializing it with NULL

void PUSH()

{ NODE *p=new NODE; **//creating new dynamic list to go on to stack**

cout<<"\nEnter Name : ";

gets(p->name);

cout<<"\nEnter Fees : ";

cin>>p->fees;

p->next=NULL;

if(top= = NULL)

top=p;

else

{ p->next=top;

top=p }

cout<<"\nList inserted on the top of stack successfully...";

getch();

}

void POP()

{ if(top= = NULL)

cout<<"\nStack Empty";

else

{ NODE *temp=top;

top=top->next;

delete temp;

cout<<"\nList deleted from top of stack successfully...";

getch();

}

}

d) Consider the class:

```
class QUEUE
```

```
{
```

```
private:
```

```
int data[20],front,rear;
```

```
public:
```

```
QUEUE( )
```

```
{ front=rear=-1; }
```

```
void INSQ(int d); //to insert an element into queue
```

```
void DELQ( ); //to delete an element from the queue
```

```
void PRINTQ( ); //to print the current status of queue
```

```
};
```

Complete the definition of function **DELQ()** of above class.

Ans.:

```
void QUEUE::DELQ( )
```

```

{
if(front<0)
cout<<"\nQueue Empty";
else
{
cout<<"\n"<<data[front]<<" has been removed from queue";
for(int i=front;i<rear;i++)
data[i]=data[i+1];
rear--;
if(rear<0)
front=-1; }
}

```

e) Evaluate the following postfix notation of expression:

2

30, 6, 4, +, /, 14, +, 4, *

SOL. (by tabular method):

Steps	INPUT	ACTION	STACK
1	30	Push	#30
2	6	Push	#30,6
3	4	Push	#30,6,4
4	+	Pop 4,6 & Push 6+4=10	#30,10
5	/	Pop 10,30 & Push 30/10=3	#3
6	14	Push	#3,14
7	+	Pop 14,3 & Push 3+14=17	#17
8	4	Push	# 17,4
9	*	Pop 4,17 & Push 17*4=68	#68

Ans. 68

4. a) Observe the program segment given below carefully and answer the question that follows :

1

```

class school
{ private :
char name[25];
int numstu;
public:
void inschool();
void outschool();
int retnumstu()
{ return numstu; }
};

void modify(school A)
{ fstream INOUT;
INOUT.open("school.dat",ios::binary|ios::in|ios::ate);
school B;
int recread=0, found=0;
while(!found && INOUT.read((char*)&B,sizeof(B))
{ recread++;

```



```

if(A.retnumstu( )= B.retnumstu( )
{
_____//missing statement

INOUT.write((char*)&A,sizeof(A));
Found=1;
}
else
INOUT.write((char*)&B,sizeof(B));
}
if(!found)
cout<<"\nRecord for modification does not exist";
INOUT.close( );
}

```

If the function **modify()** is supposed to modify a record in file **school.dat** with the values of school A passed to its argument, write the appropriate statement for missing statement using **seekp()** or **seekg()**, whichever needed, in the above code that would write the modified record at its proper place.

Ans. :

INOUT.seekp(-1*sizeof(school),ios::cur);

OR

INOUT.seekg(-1*sizeof(school),ios::cur);

b) Write a function to *count the number of vowels* stored in a text file “STRINGS.TXT”. 2

Ans.:

void countvowel()

```

{ int c=0;
char ch;
ifstream fin(“STRINGS.TXT”);
while(!fin.eof( ))
{
fin.get(ch); OR fin>>ch;
if(!fin)
break;
switch(ch)
{
case ‘A’:
case ‘a’:
case ‘E’:
case ‘e’:
case ‘I’:
case ‘i’:
case ‘O’:
case ‘o’:
case ‘U’:
case ‘u’:c++;
}
}
cout<<"\nTotal vowels in the data file is “<<c;
fin.close( ); }

```

c) Write a function to *delete* a record on the given *model number* for a *TV* from the binary file 3

“**TV.DAT**” containing the objects of **TV** (as defined below) :

```
class TV
{
long model;
float size;
char brand[30],comp[30];
public:
long retmodel()
{ return model; }
void Input() { cin>>model>>size; gets(brand); gets(comp); }
void Output() { cout<<model<<size<<brand<<comp<<endl; } };
```

Ans.:

```
void DELREC(long m)
{ TV ob;
ifstream fin("TV.DAT",ios::binary);
ofstream fout("temp.dat",ios::app|ios::binary);
int flag=0;
while(!fin.eof()) //for searching record
{ fin.read((char*)&ob,sizeof(TV));
if(!fin)
break;
if(ob.retmodel()==m)
{ flag=1;
break; }
}
if(!flag)
{ cout<<"\nRecord does not exist";
getch(); }
else //for deleting record
{ fin.seekg(0);
while(!fin.eof())
{ fin.read((char*)&ob,sizeof(TV));
if(!fin)
break;
if(ob.retmodel()==m)
fout.write((char*)&ob,sizeof(ob)); }
remove("TV.DAT");
rename("temp.dat","TV.DAT");
cout<<"\nRECORD DELETED SUCCESSFULLY.....";
getch(); }
fin.close();
fout.close();}
```

5. a) What do you understand by **Primary Key** and **Alternate Key**. Explain with example. 2

PRIMARY KEY : It is a set of one or more attributes that can uniquely identify tuples within the relation.

ALTERNATE KEY : A candidate key that is not the primary key is known as an alternate key.

For ex.

Relation: **Data**

EmpNo
Name
Designation
MobileNo
PANCardNo
Salary
BankAccountNo

Here in above table EmpNo, MobileNo, PANCardNo & BankAccountNo are candidate keys. If EmpNo is made the primary key then remaining will automatically become alternate keys.

b) Consider the following table **GAMES** and **PLAYER**. Write SQL commands for the statements (i) to (iv) and give outputs for SQL queries (v) to (viii).

Table : GAMES

GCODE	GAMENAME	NUMBER	PRZMONEY	SCHDATE
101	Chess	5	25000	23 Jan 2010
102	Badminton	3	38000	12 Nov 2008
103	Carrom	6	18000	18 Mar 2010
105	Table Tennis	3	30000	09 Jan 2009
108	Basketball	5	40000	29 Apr 2009

Table : PLAYER

PCODE	NAME	GCODE
1	Rakesh Srivastava	101
2	Nilesh Mishra	102
3	Vandana	108
4	Ravi Jindal	105

(i) to display the details of those games which are having prize money less than 30000 and organized before 2009. 1

Ans. : SELECT * FROM GAMES WHERE PRZMONEY<30000 AND SCHDATE<'01-JAN-2009' ;

(ii) to display the name of PLAYERS in reverse alphabetical order. 1

Ans. : SELECT NAME FROM PLAYERS ORDER BY NAME DESC;

(iii) to increase the prize money by 1000 for those games which name starts with 'B'. 1

Ans. : UPDATE GAMES SET PRZMONEY=PRZMONEY+1000 WHERE GAMENAME LIKE 'B%';

(iv) Insert an additional attribute namely DOB for entering date of birth in table PLAYER. 1

Ans. : ALTER TABLE PLAYER ADD(DOB DATE);

(v) SELECT GAMENAME,NAME FROM GAMES G,PLAYER P WHERE G.GCODE=P.GCODE; ½

Ans. :

GAMENAME	NAME
-----	-----
Chess	Rakesh Srivastava
Badminton	Niesh Mishra
Basketball	Vandana
Table Tennis	Ravi Jindal

(vi) SELECT MIN(SCHDATE), MAX(PRZMONEY) FROM GAMES ; ½

Ans. :

MIN(SCHDATE)	MAX(PRZMONEY)
-----	-----
12-Nov-2008	40000

(vii) SELECT AVG(PRZMONEY) FROM GAMES WHERE SCHDATE<'01-JAN-2009'; ½
Ans. : **AVG(PRZMONEY)**

38000

(viii) SELECT COUNT(DISTINCT NUMBER) FROM GAMES; ½
Ans. : **COUNT(DISTINCT NUMBER)**

3

6. a) State and verify *Absorption law* in Boolean algebra. 2

Ans.: Absorption law states that :

a) $x+xy=x$ b) $x(x+y)=x$

Verification:

$x+xy=x$

LHS= $x+xy$ by distributive law

= $x(1+y)$ since $1+y=1$

= $x.1$ since $1.x=x$

= x = RHS, hence verified

OR (using truth table)

x	y	xy	x+xy
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

Here column x and x+xy are identical, hence proved.

b) Write the SOP form of a Boolean function G, which is represented in a truth table as follows : 2

A	B	C	G
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Ans. Add extra column for min term and write min terms for the rows which have output (G) as 1:-

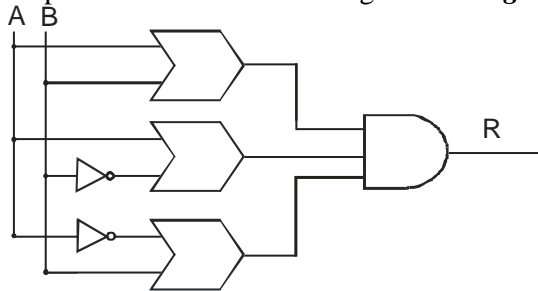
A	B	C	G	Min Term
0	0	0	1	$A'B'C'$
0	0	1	1	$A'B'C$
0	1	0	0	
0	1	1	0	
1	0	0	1	$AB'C'$
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

Now sum all the min terms to get the SOP as:

Therefore SOP of F(A,B,C)= $A'B'C'+A'B'C+AB'C'+ABC$

c) Write the equivalent Boolean Expression R for the following *circuit diagram* :

1



Ans. : $(A+B) (A+B') (A'+B)$

d) If $F(P,Q,R,S) = \pi (0,2,4,5,6,7,8,10,11,12,14)$, obtain the simplified form using *K-Map*.

3

Ans. Draw the 4 variable K-Map, plot & group the 0s starting from bigger to smaller group:

	RS			
PQ	0+0	0+1	1+1	1+0
0+0	0 0	1	3	2 0
0+1	0 4	0 5	0 7	0 6
1+1	0 12	13	15	0 14
1+0	0 8	9	0 11	0 10

Reducing:

Octet = M0.M2.M4.M6.M8.M10.M12.M14
= S

Quad = M4.M5.M6.M7
= $P+Q'$

Pair = M10.M11
= $P'+Q+R'$

Therefore POS of $F(P,Q,R,S) = S.(P+Q').(P'+Q+R')$

7. a) What is the difference between *packet & message* switching?

1

Ans.: Packet Switching: It refers to protocols in which messages are broken up into small packets before they are sent. Each packet is transmitted individually across the net. Each packet has header information which enables to route the packet to its destination. At the destination the packets are reassembled into the original message.

Message Switching: In this technique, first the source computer transfers data to the buffer of switching office computer. Further it looks for a free link to another switching office, and then the data are transferred to this link.

b) Expand the following terminologies :

1

i) PHP

ii) SMSC

Ans.: i) PHP – Hypertext Preprocessor

ii) SMSC – Short Message Service Center

c) What is infrared technology?

1

Ans.: INFRARED : Infrared technology allows computing devices to communicate via short-range wireless signals. The infrared transmission technology used in computers is similar to that used in consumer product remote control units. This ray transmits digital data bi-directionally through the air and can propagate throughout a room, but will not penetrate walls.

d) What do you mean by *spam*?

1

Ans.: SPAM :- It refers to electronic junk mail or junk newsgroup postings. Some people define it as any unsolicited e-mail.

e) What is *proprietary software*?

1

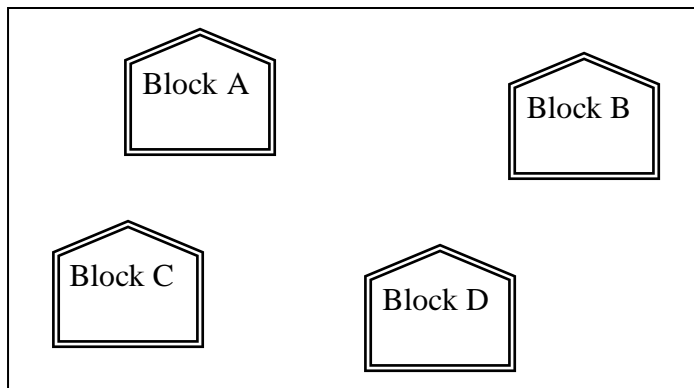
Ans.: Proprietary software: It refers to any computer software that has restrictions on any combination of the usage, modification, copying or distributing modified versions of the software. Proprietary software usually can be distributed at no cost or for a fee. Proprietary software may also be called closed-source software. In other words it is neither open nor freely available.

f) What is *Web Hosting*?

1

Ans.: Web hosting: It is a way of hosting web-server application on a computer system through which electronic content on the internet is readily available to any web-browser client.

g) The Rangoli Creation has set up its new center at Patna for its office & web based activities. It has four blocks of buildings as shown in the diagram below:



The distance between various blocks are :

Block A to Block B	30 m
Block B to Block C	110 m
Block C to Block D	55 m
Block A to Block D	260 m
Block B to Block D	195 m
Block A to Block C	32 m

Number of computers in each block are :

Block A	25
Block B	55
Block C	125
Block D	15

(g1) Suggest the cable layout (with diagram) of connections among the blocks & technology.

1

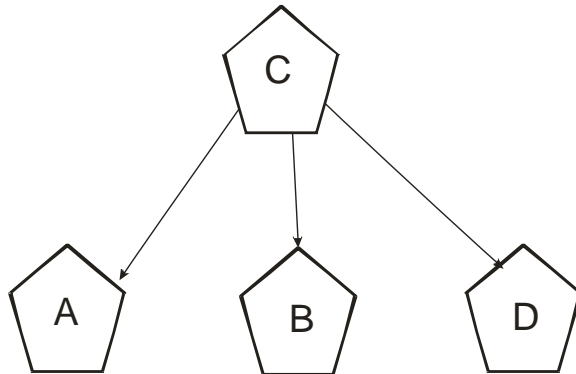
Ans. : (Draw any one)

1

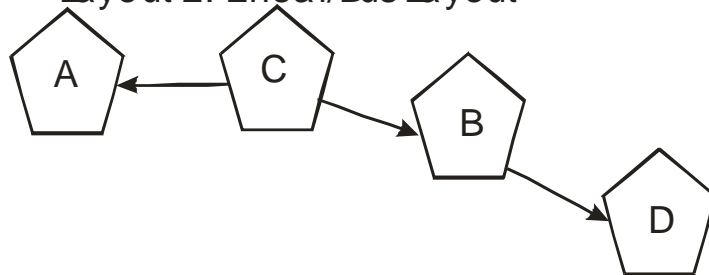
1

1

Layout 1: Star Layout



Layout 2: Linear/Bus Layout



(g2) Suggest the most suitable place to house the server, with a suitable reason.

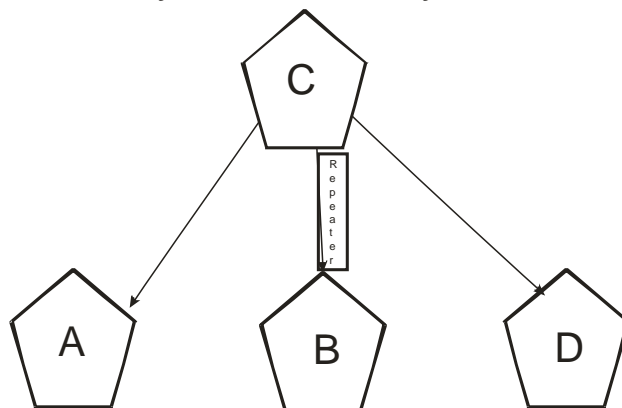
Ans.: The most suitable place to house the server would be Block C as it has the maximum number of computers.

(g3) Suggest the placement of the following devices with reasons :

- i) Repeater
- ii) Switch/Hub

Ans.: i) In layout 1 repeater will be placed between C & B blocks

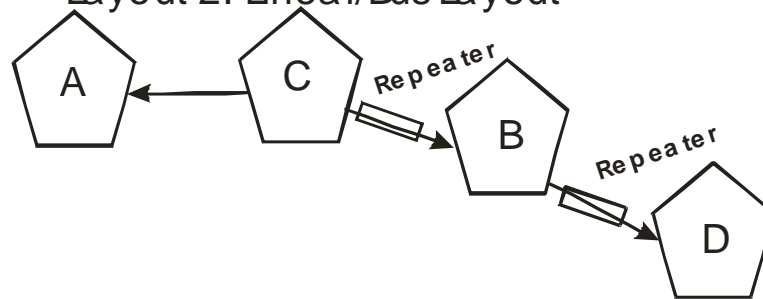
Layout 1: Star Layout



OR

In layout 2, repeater will be placed between C & B and between B & D blocks.

Layout 2: Linear/Bus Layout



ii) *Switch/Hub* will be placed in all the blocks as they have their own computer networks.

(g4) The organization is planning to link its another office in the city located in the hilly region where cable connection is not feasible. Suggest an economic way to connect it with reasonably high speed. Justify your answer.

Ans.: Radio Wave would be an economic way to connect it with reasonably high speed. It offers mobility & freedom from land acquisition rights that are required for laying, repairing cables.