

June : 2016 (CBCS)

Note :

Max. marks : 60

- (i) Attempt any five questions.
- (ii) All questions carry equal marks.

**Q.1 (a)** What do you understand by the instruction set? What is the step taken by the CPU to execute the instructions?

**Ans. Instruction set :** The computer system cannot do anything on its own. They are programmed or instructed by the user. The instruction is defined as a machine language command to perform a specific task. It should be understandable by the computer system, hence it is a string of binary digits (0 and 1).

**Example :** It may be issued to read data from the memory, write data into the memory, etc. Generally, an instruction is used to perform a single task.

The set of instructions used to perform specific operations is called the **instruction set**. Each instruction has two parts : **opcode** and **operand**.

1. **Opcode :** It specifies the operation to be performed by the instruction issued to the computer system such as ADD to perform addition, MOV to move data from one location to another.
2. **Operand :** It specifies the memory locations where the input and output data are kept.

The instructions are classified into three categories according to the number of bytes required for their execution. These are as follows :

1. **One-byte instructions :** The instructions that require only one byte for their execution.

**Example :** MOV

2. **Two-byte instructions :** The instructions that require one byte for their execution and one byte for their operand.

**Example :** MVI

3. **Three-byte instructions :** The instructions that require one byte for their execution and two bytes for their operands. These instructions are used to specify the 16 bit operand with the instruction.

**Example :** STA

According to the number of instructions required and their complexity, based on these instruction sets, the computer architecture is also categorized into two categories : CISC and RISC.

1. **CISC :** It stands for **complex instruction set computer**. It provides a large number of instructions. These instructions include some complex instructions to perform complex tasks. These were developed to complete the complicated task in an easy and flexible manner, so that the compiler or other translator has to do very little work in translating the code written in some programming language into machine language. These instructions are memory-based, and the computer requires a separate circuitry for these instructions. Therefore, more time is required for their execution and their design is complex.

**Example :** MULT, JA, JPO.

2. **RISC :** It stands for **reduced instruction set computer**. It is based on the concept that complex operation can also be performed using simple instruction. For example, the multiplication can be performed by loading the contents in memory; multiplying and then storing them back into the memory. These are the three simpler steps to perform the multiplication, instead of performing it in a single step. Therefore, the RISC computer uses sets of simple instructions that may be completed into a single cycle. Also, it is faster and less expensive than CISC.

**Example :** ADD, COMPARE.

The processing of an instruction by the CPU consists of following steps :

1. **Fetching** : The CPU fetch the instruction from the memory, and the program counter is replaced by the next instruction.
2. **Decoding** : The instruction is decoded by the CPU (CU and ALU) to determine the action to be performed.
3. **Execute** : The instruction is executed to perform a specific task.
4. **Store** : The result generated by the execution of an instruction is stored back into the memory.

The completion of all the steps for an instruction is called an instruction cycle of the machine cycle. The instruction cycle is shown in figure below :

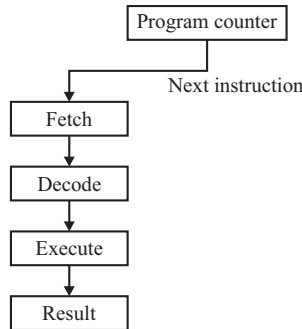


Fig. : Instruction cycle

**Q.1 (b)** Differentiate RAM and ROM. Explain the Dynamic and Static RAM with their merits and demerits.

**Ans. Difference between RAM and ROM :**

S. No.	Criteria	RAM	ROM
1.	<b>Stands for</b>	Random access memory. It is also called as volatile memory.	Read only memory. It is also called as non-volatile memory.
2.	<b>Definition</b>	Random access memory or RAM is a form of data storage medium that can be accessed randomly at any time.	Read-only memory or ROM is a form of data storage medium that permanently stores data on personal computers and other electronic devices.
3.	<b>Accessibility</b>	The information stored in the RAM is easily accessed by the processor.	The processor cannot directly access the information that is stored in the ROM.
4.	<b>Working type</b>	Both the read and write operations can be performed in RAM.	The ROM memory only allows the user to read the information. User cannot make any changes to the information.
5.	<b>Storage</b>	RAM memory is only used to store the temporary information.	ROM memory is used to store permanent information.

S. No.	Criteria	RAM	ROM
6.	<b>Speed</b>	The accessing speed of RAM is faster.	The accessing speed of ROM is slower as compared to RAM.
8.	<b>Cost</b>	The price of RAMs are comparatively high.	The price of ROMs are comparatively low.
7	<b>Structure</b>	RAM is available in the form of chip.	ROM is available in the form of optical drivers that are made of magnetic tapes.
9.	<b>Chip size</b>	Physically size of RAM chip is larger than ROM chip	Physically size of ROM chip is smaller than RAM chip.
10.	<b>Types</b>	The RAM is categorized into two types :  (i) Static RAM (SRAM).  (ii) Dynamic RAM (DRAM).	The ROM is categorized into three types :  (i) PROM (Programmable read only memory).  (ii) EPROM (Erasable programmable read only memory).  (iii) EEPROM (Electrically erasable programmable read only memory)

**Advantage of static RAM :**

1. Static RAMs are faster.
2. Static RAMs are less complex.
3. Static RAMs do not require refreshing operation at regular interval.
4. Static RAMs are directly used in cache memory.

**Disadvantages of SRAM :**

1. Static RAMs are expensive.
2. Static RAMs requires more space.
3. Static RAMs consumes more power.

**Advantages of DRAM :**

1. Dynamic RAMs are simple.
2. Dynamic RAMs require less space.
3. Dynamic RAMs consumes less power.
4. Dynamic RAMs are cheaper.

**Disadvantages of DRAM :**

1. Dynamic RAMs are slower than static RAMs in speed.
2. Dynamic RAMs also require refreshing operation after regular intervals.
3. Dynamic RAMs cannot be used where only small amount of memory is needed.

Q.2 (a) What is Software? Explain system and application software. Enlist system and application software.

**Ans. Computer software :** Computer software is the set of programs that makes the hardware perform a set of tasks in particular order. Hardware and software are complimentary to each other. Both have to work together to produce meaningful results. Computer software is classified into two broad categories, system software and application software.

1. **System software :** System software consists of a group of programs that control the operations of a computer equipment including functions like managing memory, managing peripherals, loading, storing, and is an interface between the application programs and the computer. MS DOS (Microsoft's disk operating system), UNIX are examples of system software.

The system softwares are divided into 3 types. They are :

(i) Operating system software      (ii) Language translator      (iii) Utility software

(i) **Operating system software (OS) :** An **operating system** is a program designed to run other programs on a computer. The various types of operating system are :

- General purpose operating system
- Special purpose operating system
- Batch processing operating system
- Multi-user operating system
- Multi-programming operating system
- Real time operating system
- Embedded operating system

(ii) **Language translator :** It is another system software which convert the high level language to machine level language for the purpose of machine understanding. There are 3 types of language translator, they are as follows :

- **Compiler :** A compiler is a program that translates a source program written in high-level programming language into machine code.
- **Interpreter :** An interpreter is a program that can analyze and execute a program line by line.
- **Assembler :** An assembler is a program for converting instructions written in low-level symbolic code into machine code.

(iii) **Utility software : Utility software** is a kind of system software designed to help, analyze, configure, optimize and maintain the computer. A single piece of utility software is usually called a utility or tool. Some of the utility software are :

- Antivirus
- Memory tester

2. **Application software :** Software that can perform a specific task for the user, such as word processing, accounting, budgeting or payroll, fall under the category of application software. Word processors, spreadsheets, database management systems are all examples of general purpose application software.

**Types of application software are :**

(a) **Word processing software :** The main purpose of this software is to produce documents. MS-Word, Word Pad, Notepad and some other text editors are some of the examples of word processing software.

(b) **Database software :** Database is a collection of related data. The purpose of this software is to organize and manage data. The advantage of this software is that you can change the way data is stored and displayed. MS access, dBase, FoxPro, paradox, and oracle are some of the examples of database software.

- (c) **Spread sheet software** : The spread sheet software is used to maintain budget, financial statements, grade sheets, and sales records. The purpose of this software is organizing numbers. It also allows the users to perform simple or complex calculations on the numbers entered in rows and columns. MS-excel is one of the examples of spreadsheet software.
- (d) **Presentation software** : This software is used to display the information in the form of slide show. The three main functions of presentation software is editing that allows insertion and formatting of text, including graphics in the text and executing the slide shows. The best example for this type of application software is Microsoft PowerPoint.
- (e) **Multimedia software** : Media players and real players are the examples of multimedia software. This software will allow the user to create audio and videos. The different forms of multimedia software are audio converters, players, burners, video encoders and decoders.

**Q.2 (b)** Write an Algorithms to find largest number from array of  $n$  numbers.

**Ans.** Suppose an array DATA having N numeric value is given. This algorithm finds the location as LOC and the largest number of the array DATA. To find the largest number we need compare the numbers one by one and for that we need an increment say k. Variable MAX is used to store the largest number.

Step 1 : Start

Step 2 : Read : N inputs for DATA array

Step 3 : [Initialization] Set LOC := 1 and MAX := DATA[1]

Step 4 : Repeat for loop from k = 1 to k ≤ N and increment k by 1

If MAX < DATA[k], then

Set LOC := k and MAX := DATA[k].

[End of If structure]

[End of for loop]

Step 5 : Print : LOC (Location of largest number) and MAX(Value of the largest number)

Step 6 : Exit

**Q.3 (a)** What is Programming Language? Compare Assembly Language and High-Level Language.

**Ans. Programming language** : A programming language is a set of words, symbols and codes that enables humans to communicate with computers. It is a language used for writing computer programs that direct a computer to perform computation and to organize the flow of control between mechanical devices.

**Difference between machine language, assembly language and high level language are :**

S.No.	Criteria	Machine language	Assembly language	High level language
1.	<b>Generation</b>	It is called as first generation language or 1GL.	It is called as second generation language or 2GL.	It is called as third generation language or 3GL.
2.	<b>Instructions</b>	It uses binary coded instructions.	It uses mnemonics for machine instructions.	It uses simple english instructions.
3.	<b>Understandable</b>	It is difficult to understand.	It is difficult to understand.	It is easier to understand.
4.	<b>Use of translator</b>	It does not require translator.	Assemblers are used to translate assembly programs.	Compilers and interpreters are used to translate high level programs.
5.	<b>Machine dependency</b>	It is machine dependent.	It is machine dependent.	It is machine independent.

S.No.	Criteria	Machine language	Assembly language	High level language
6.	<b>Knowledge of hardware</b>	The programmer needs the knowledge of hardware.	The programmer needs the knowledge of hardware.	The programmer does not need the knowledge of hardware.
7.	<b>Execution speed</b>	It is executed very fast as compared to assembly language and high level language.	It is executed fast as compared to high level language.	It is executed slow as compared to machine language and assembly language.

**Q.3 (b)** Explain the different control statements in C++.

**Ans.** C++ provides two types of control structures :

1. Decision making statements
2. Loop control statements

**1. Decision making statements :**

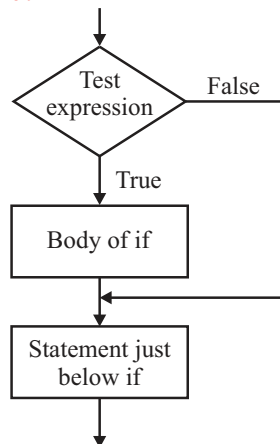
(i) **if statement :** The if, if...else, nested if...else statement and else if ladder are used to make one-time decisions in C++ Programming, that is, to execute some codes and ignore some codes depending upon the test expression.

**if (test expression)**

```
{
statements      to be executed if test expression is true;
}
```

The if statement checks whether the test expression inside parenthesis ( ) is true or not. If the test expression is true, statements inside the body of if statement is executed but if test is false, statements inside body of if is ignored.

**Flowchart of if statement :**



**Fig. : Flowchart of if statement**

(ii) **if...else statement :** The if...else statement is used if the programmer wants to execute some statements when the test expression is true and execute some other statements if the test expression is false.

**Syntax of if...else :**

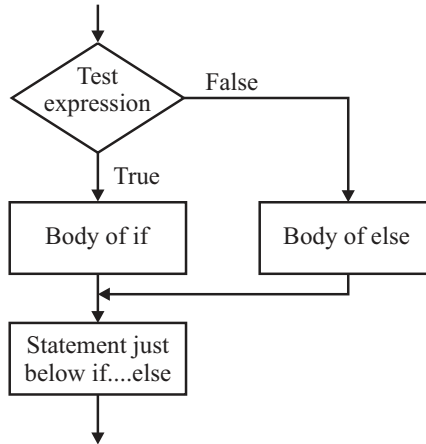
**if (test expression)**

```
{
statements to be executed if test expression is true;
}
```

```

else
{
    statements to be executed if test expression is false;
}
    
```

**Flowchart of if...else statement :**



**Fig. : Flowchart of if...else statement**

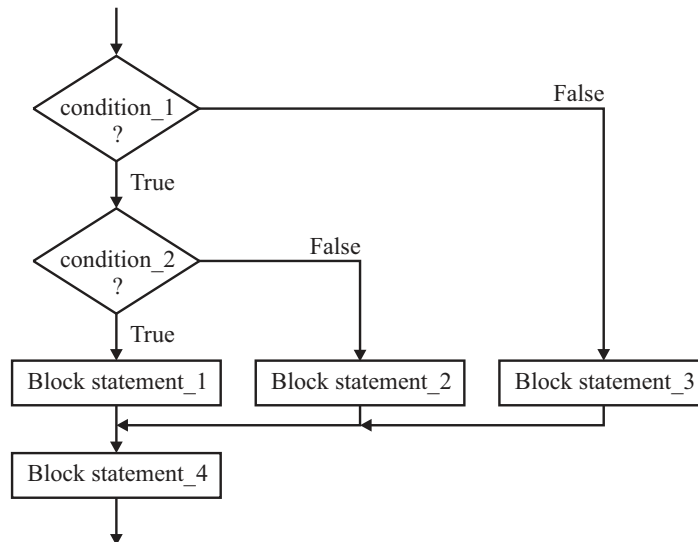
**(iii) Nested if...else statement :** The nested if...else statement is used when there are two or more conditions to be satisfied but the checking of one condition is dependent on another condition and if both conditions are found to be true, then the decision is to be executed. In this statements, the if statements are nested together and the decisions are executed after checking of various conditions.

**Syntax of nested if...else statement :**

```

if (test expression1)
{
    if(test expression2)
    {
        statements to be executed if test expression1 and 2 is true;
    }
}
    
```

**Flowchart of nested if...else statement :**



**Fig. : Nested if...else statement**

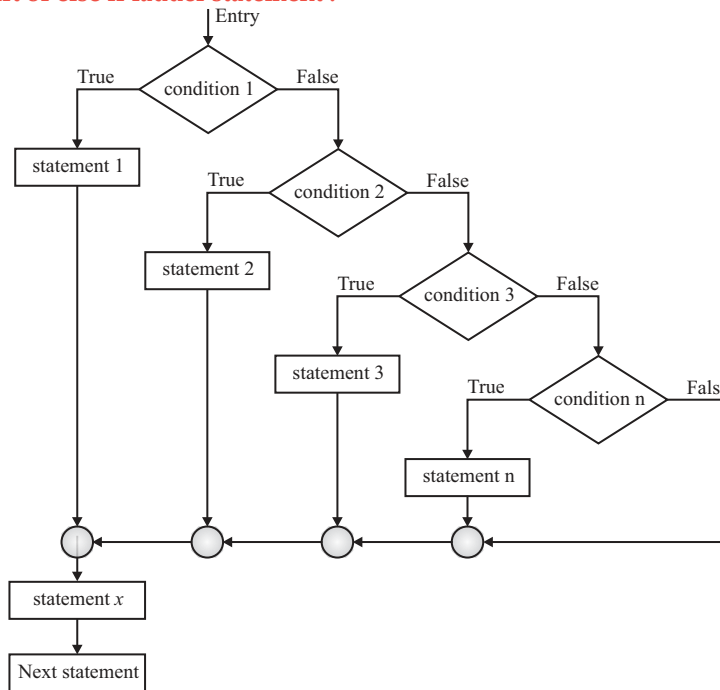
(iv) **else if ladder statement (if...else if....else statement)** : The else if ladder statement is used when program requires more than one test expression. The American National Standard Institute (ANSI) standard specifies that 15 levels of nesting may be continued.

**Syntax of else if ladder statement :**

```

if (test expression1)
{
    statements to be executed if test expression1 is true;
}
else if(test expression2)
{
    statements to be executed if test expression1 is false and 2 is true;
}
else if (test expression 3)
{
    statements to be executed if test expression1 and 2 are false and 3 is true;
}
.
.
.
else
{
    statements to be executed if all test expressions are false;
}
    
```

**Flowchart of else if ladder statement :**



**Fig. : else if ladder statement**

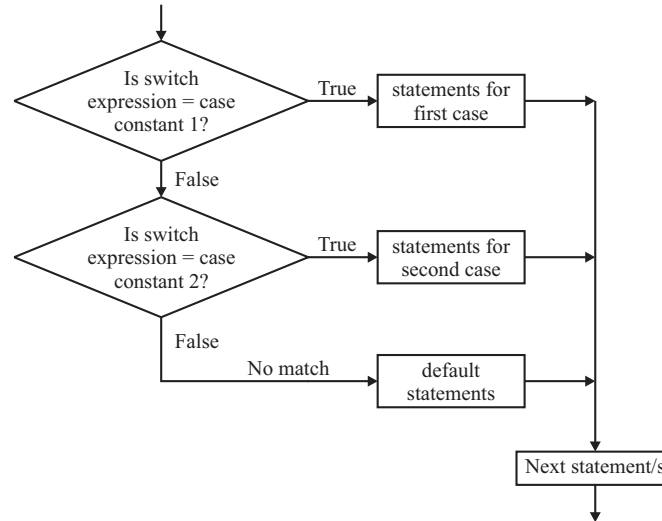
(v) **switch case statement** : Decision making are needed when, the program encounters the situation to choose a particular statement among many statements. If a programmer has to choose one block of statement among many alternatives, nested if...else can be used but, this makes programming logic complex. This type of problem can be handled in C++ programming using switch statement.



**Syntax of switch...case :**

```
switch (n)
{
case constant1:
    statements to be executed if n equals to constant1;
break;
case constant2:
    statements to be executed if n equals to constant2;
break;
.
.
.
default:
    statements to be executed if n doesn't match to any cases;
}
```

The value of n is either an integer or a character in above syntax. If the value of n matches constant in case, the relevant codes are executed and control moves out of the switch statement. If the n doesn't match any of the constant in case, then the default codes are executed and control moves out of switch statement.

**Flowchart of switch...case statement :****Fig. : switch case statement**

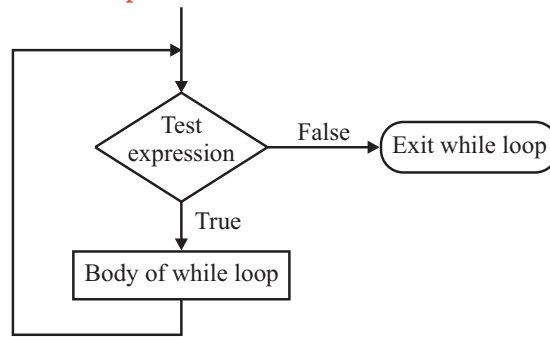
The break statement at the end of each case cause switch statement to exit. If break statement is not used, all statements below that case statement are also executed.

**Role of break and default statements in switch :**

**break :** We can use the **break** statement to end processing of a particular case within the switch statement and to branch to the end of the switch statement. Without **break**, the program continues to the next case, executing the statements until a **break** or the end of the statement is reached. In some situations, this continuation may be desirable.

**default :** The **default** statement is executed if no **case constant-expression** is equal to the value of **switch expression**. If the **default** statement is omitted, and no **case** match is found, none of the statements in the switch body are executed. There can be at most one **default** statement. The **default** statement need not come at the end, it can appear anywhere in the body of the switch statement. A **case** or **default** label can only appear inside a switch statement.



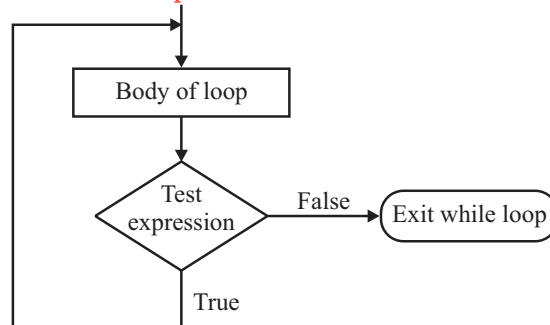
**Flowchart of while loop :****Fig. : while loop**

(iii) **do...while loop** : In C++, do...while loop is very similar to while loop. Only difference between these two loops is that, in while loops, test expression is checked at first but, in do...while loop code is executed at first then the condition is checked. So, the code are executed at least once in do...while loops.

**Syntax of do...while loops :**

```
do  
{  
    some codes;  
} while (test expression);
```

At first codes inside body of do is executed. Then, the test expression is checked. If it is true, codes inside body of do are executed again and the process continues until test expression becomes false (zero). There is semicolon in the end of while in do...while loop.

**Flowchart of do...while loop :****Fig. : do...while loop**

**Q.4 (a)** What is Inheritance? Write the different types of Inheritance.

**Ans. Inheritance** : The mechanism of creating new class from an old class is called **inheritance**. The old class is known as **base class** and new class is known as **derived class**. The derived class inherits all the properties from the base class.

**Type of inheritance :**

1. Single inheritance
2. Multiple inheritance
3. Hierarchical inheritance
4. Multilevel inheritance
5. Hybrid inheritance

**1. Single inheritance** : When a derived class inherits the properties from only one base class, it is known as single inheritance.

**Example** : Consider a derived class B which inherits the properties from base class A.

**Syntax for single inheritance :**

```
class A
{
};
class B : public A
{
};
```

2. **Multiple inheritance :** When a derived class inherits the properties from more than one class, it is known as multiple inheritance.

**Example :** Consider a derived class C which inherit the properties from base classes A and B.

**Syntax for multiple inheritance :**

```
class A
{
};
class B
{
};
class C : public A, public B
{
};
```

3. **Hierarchical inheritance :** When several derived classes inherits the properties from one base class, it is known as hierarchical inheritance.

**Example :** Consider two derived classes B and C which inherits the properties from base class A.

**Syntax for hierarchical inheritance :**

```
class A
{
};
class B : public A
{
};
class C : public A
{
};
```

4. **Multilevel inheritance :** When a derived class inherits the properties from another derived class it is known as multilevel inheritance.

**Example :** Consider a derived class C which inherit the properties from derived class B which in turn inherits the properties from base class A.

**Syntax for multilevel inheritance :**

```
class A
{
};
class B : public A
{
};
class C : public B
{
};
```

5. **Hybrid inheritance :** When we mix any two types of inheritances in our program, it is known as hybrid inheritance.

**Example :** Consider a derived class C which inherit the properties from base classes A and B.

**Syntax for hybrid inheritance :**

```
class A
{
};
class B : public A
{
};
class C
{
};
class D : public B, public C
{
};
```

**Q.4 (b)** What is a constructors? What is the importance of a constructors?

**Ans. Constructor :** A constructor is a special member function whose task is to initialize the objects of its class. It is special because its name is the same as the class name. The constructor is invoked whenever an object of its associated class is created. It is called constructor because it constructs the values of data members of the class.

**A constructor is declared and defined as follows :**

```
// class with a constructor
class integer
{
    int m, n;
public:
    integer(void);           // constructor declared
    .....
    .....
};
integer :: integer(void)   // constructor defined
{
    m = 0;
    n = 0;
}
```

When a class contains a constructor like the one defined above, it is guaranteed that an object created by the class will be initialized automatically.

**Example :** The declaration

```
integer int1;           // object int1 created
```

not only creates the object int1 of type integer but also initializes its data members m and n to zero. There is no need to write any statement to invoke the constructor function (as we do with the normal member functions). If a 'normal' member function for zero initialization, we would need to invoke this function for each of the objects separately. This would be very inconvenient, if there are a large number of objects.

**Types of constructors :**

1. Default constructor
2. Parameterized constructor
3. Copy constructor

1. **Default constructor** : A constructor that accepts no parameters is called as the default constructor. The default constructor for **class A** is **A::A()**. If no such constructor is defined, then the compiler supplies a default constructor. Therefore a statement such as

```
A a;
```

invokes the default constructor of the compiler to create the object a.

2. **Parameterized constructor** : The constructors that can take arguments are called as parameterized constructors. These constructors are used to initialize the various data members of different objects with different values when they are created. The parameterized constructor for **class A** is **A::A (int m)**. Therefore a statement such as

```
A a(10); // implicit call
```

Or 

```
A a = A (10); // explicit call
```

invokes the parameterized constructor to create the object a.

3. **Copy constructor** : Copy constructor is used to create a new object as a copy of an existing object. The argument of such a constructor is a reference to an object of the same type as is being constructed. The compiler automatically creates a copy constructor for each class but according to the requirements the programmer has to create the copy constructor, known as a user-defined copy constructor. The copy constructor for **class A** is **A::A (A &m)**. Therefore a statement such as

```
A b(a); // implicit call
```

Or 

```
A b = A (a); // explicit call
```

invokes the parameterized constructor to create the object b.

**Example** : Program to show the use of constructors.

```
#include<iostream>
using namespace std;
class Sample
{
    int a;
    public :
        Sample() // Default Constructor
        {
            a = 10;
        }
        Sample(int b) // Parameterized Constructor
        {
            a = b;
        }
        Sample(Sample &m) //Copy Constructor
        {
            a=m.a;
        }
        void show()
        {
            cout<<"na ="<<a;
        }
};
int main()
{
    Sample s1; //Default Constructor called
    Sample s2(100); //Parameterized Constructor called
    Sample s3 = s1; //Copy Constructor called
    s1.show();
}
```

```
s2.show();  
s3.show();  
return 0;  
}
```

**Output :**

```
a = 10  
a = 100  
a = 100
```

**Q.5 (a)** What is E-Commerce? What are the advantages and disadvantages of E-Commerce?

**Ans. E-commerce :** E-commerce stands for **electronic commerce**. It is also known as e-trade or e-business. E-commerce is the term used to describe the selling of goods and services over the Internet. This saves time for users on both ends. The users can buy, sell, and exchange products or services via computer network. Usually the payments are made using credit cards.

**Role of e-commerce in networking :** The main services provided by e-commerce are as follows :

1. **E-mail :** E-mail is the most popular service through which we can electronically send and receive messages anywhere in the world.
2. **Video conferencing :** Video conferencing is a meeting between two or more people located at different places. It is conducted through computer network. It provides an environment of normal meeting. It enables participants to see and hear each other at the same time as if they are in the same room.
3. **Electronic shopping (E-shopping) :** The shopping conducted through internet is known as electronic shopping or e-shopping. Many business organizations have their websites on the internet. These websites are used to sell goods and services. Customers place their orders through websites and make payments using credit cards.
4. **Electronic banking :** A service that allows an account holder to obtain account information and manage certain banking transactions through computer network is called electronic banking or e-banking. An electronic banking is also known as online banking or cyber-banking.

**Advantages of e-commerce :**

1. **Low cost :** The entire financial sector will eventually become electronic, so a sooner conversion is going to be lower on cost; thus, also lowering operational costs as no physical needs to be set up. It makes every transaction through e-commerce payment a lot cheaper.
2. **Higher profit margin, better productivity :** E-commerce also enables us to move better with higher margin for more business safety. Higher margin also means business with more control as well as flexibility. One can also save time from the e-commerce as it is all time available. Productivity here means productivity for both companies and customers.
3. **Effectiveness :** Customers can easily select products from different providers without moving around physically.
4. **Instant comparison :** E-commerce also enables to compare price among several providers. In the end, it leads us to smart shopping. People can save more money while they shop.
5. **Economy benefit :** E-commerce allows us to make transaction without any needs on stores, infrastructure investment, and other common things we find. Companies only need well built website and customer service. Thus it is easy to start and manage a business.

**Disadvantages of e-commerce :**

1. **Security :** Customers need to be confident and trust the provider of payment method. Sometimes, we can be tricked. Examine on integrity and reputation of the web stores before you decide to buy.

- Scalability of system :** A company definitely needs a well developed website to support numbers of customers at a time. If the web destination is not well enough it is better not to use it.
- Integrity on data and system :** Customers need secure access all the time. In addition to it, protection to data is also essential. Unless the transaction can provide it, we should refuse for e-commerce. Such sites are soft targets for hackers.
- The “touch and feel” people :** People who prefer and focus on product will not buy online. They will want to feel, try, and sit on their new couch and bed. They believe there is no guarantee of product quality online.
- Customer service and relation problem :** They sometimes forget how essential to build loyal relationship with customers. Without loyalty from customers, they will not survive the business. As there is minimum chance of direct customer to company interactions, customer loyalty is always on a check.

**Q.5 (b)** Write the various networking devices.

**Ans.** Networks are connected by using some types of devices such as hubs, repeaters, switches, bridges, routers and gateways. It is very important to know the basic function of these devices in order to decide upon the device that is to be used for a particular purpose.

- Hub :** A hub works in the physical layer of the OSI model. Hub is a non-intelligent device, and has no decision making capability. Hub basically takes the input data from one of the ports and broadcast the information to all the other ports connected to the network.

**There are two types of hub :**

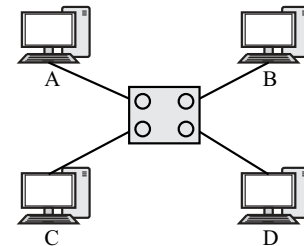
- Active hub :** Active hub regenerate, concentrate and strengthen the signals before sending them to their destinations.
  - Passive hub :** Passive hub simple broadcast the signals.
- Repeater :** A repeater is a device similar to the hub, but has additional features. It also works in the physical layer. The repeaters are used in places where amplification of input signal is necessary. It regenerates the input signal, and amplifies only the desirable signal. Hence, the noise component of the signal is eliminated. It is also called as **active hub**.

The repeaters are useful for long distance transmission.

One common problem between the repeaters and the hubs are that only one transmission can take place on the network at a particular time. If multiple devices transmit data simultaneously, there will be data collision.

- Switch :** A switch is an intelligent device that works in the data link layer. The term intelligent refers to the decision making capacity of the switch. It has knowledge of the MAC addresses of the ports in the network.

It establishes a link between the sender and the receiver based on the MAC addresses. Simultaneous data transfer is possible in a switch. Switch does not divide bandwidth. A switch is a secure device, because it sends information only to the desired destinations, and also certain security features such as firewalls can be implemented in the switches.



**Fig. : Port network (Hub)**



**Fig. : Repeater**



**Fig. : Switch**

and also certain security features such as firewalls can be implemented in the switches.



4. **Bridge** : A bridge is also a device which works in the data link layer, but is more primitive when compared to a switch. Initial bridges were used to connect only 2 LAN's, but the most recent ones perform similar operation as the switches. It also works on the principle of transfer of information using the MAC addresses of the ports.

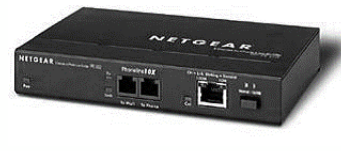


Fig. : Bridge

When bridging is used, each time the device has to be connected to the internet. A bridge alone cannot be used to connect to the internet. It has no knowledge of the IP addresses used in the internet.

5. **Router** : Router connects two different LANs. Router works in the network layer. Router send the on the basis of IP addresses. Routers are similar to bridges but there are some additional features, which routers provide such as ability to filter messages and forward them to different location based on various criteria. Routers are used to find the best route for data packets to travel in the network. The process of finding the path is called as routing. The information about the best route is stored in the table called as routing table.

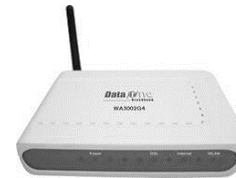


Fig. : Router

**There are two types of routers :**

- (i) **Static router** : In static router, users need to specify the routing path.  
(ii) **Dynamic router** : Dynamic router determine the routing path automatically by using different algorithms.
6. **Gateway** : A gateway is a software or hardware device is used to connect two or more networks of dissimilar types. The gateways are useful for communication between different network architecture with different protocol. Gateway is also called as protocol converter. Gateway accept the packet formatted for one protocol and convert it to a packet formatted for another protocol.

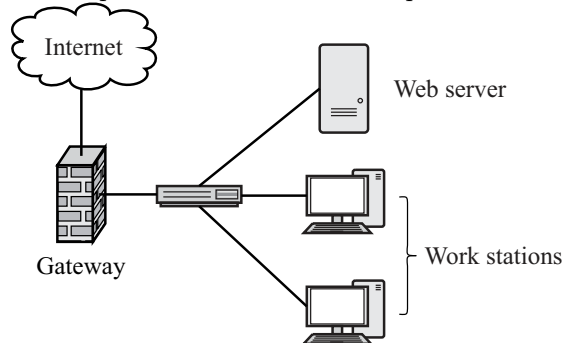


Fig. : Gateway

7. **Network interface card (NIC)** : A network interface card (NIC) is a computer hardware component that allows a computer to connect to a network. NICs may be used for both wired and wireless connections. A NIC is also known as a network interface controller (NIC), network interface controller card, expansion card, computer circuit board, network card, LAN card, network adapter or network adapter card (NAC).



Fig. : Network Interface Card

**Q.6 (a)** What is DBA? Write any two responsibilities of DBA.

**Ans. Database administrator (DBA) :** The database administrator is a person having central control over data and programs accessing that data. He coordinates all the activities of the database system. The database administrator has a good understanding of the enterprise's information resources and needs.

**Functions of a DBA :**

1. **Schema definition :** The creation of the original database schema. This involves writing a set of definitions in a DDL (data storage and definition language), compiled by the DDL compiler into a set of tables stored in the data dictionary.
2. **Storage structure and access method definition :** Writing a set of definitions translated by the data storage and definition language compiler.
3. **Schema and physical organization modification :** Writing a set of definitions used by the DDL compiler to generate modifications to appropriate internal system tables (Example : Data dictionary). This is done rarely, but sometimes the database schema or physical organization must be modified.
4. **Granting user authority to access the database :** Granting different types of authorization for data access to various users.
5. **Specifying integrity constraints :** Generating integrity constraints. These are consulted by the database manager module whenever updates occur.
6. **Routine maintenance :** It includes the following :
  - (i) Acting as liaison with users.
  - (ii) Monitoring performance and responding to changes in requirements.
  - (iii) Periodically backing up the database.

**Q.6 (b)** Explain DDL commands with the help of example.

**Ans. Data definition language (DDL) :** The data definition language (DDL) is used to create and destroy databases and database objects. These commands will primarily be used by database administrators during the setup and removal phases of a database project. The five basic DDL commands :

1. CREATE
2. ALTER
3. DROP
4. RENAME
5. TRUNCATE

**1. CREATE :**

**(a) Create table :** This command is used to create a new relation.

**Syntax :**

```
Create table relation_name  
(field_1 data_type(Size),  
field_2 data_type(Size), ... );
```

**Example :**

```
SQL>create table Student (sno NUMBER(3),sname char(10),class char(5));
```

**(b) create table..as select.... :** This is used to create the structure of a new relation from the structure of an existing relation.

**Syntax :**

```
Create table (relation_name_1, field_1,field_2,.....field_n) AS SELECT field_1, field_2,  
.....field_n from relation_name_2;
```

**Example :**

```
SQL>create table std(rno,sname) as select sno,sname from student;
```

**2. ALTER :**

(a) **ALTER TABLE...ADD...:** This is used to add some extra fields into existing relation.

**Syntax :**

```
ALTER TABLE relation_name ADD (new field_1 data_type(size), new field_2 data_type(size),...);
```

**Example :**

```
SQL>ALTER TABLE std ADD(Address CHAR(10));
```

(b) **ALTER table...modify...:** This is used to change the width as well as data type of fields of existing relations.

**Syntax :**

```
Alter table relation_name modify (field_1 newdata_type(Size), field_2 newdata_type(Size),....field_newdata_type(Size));
```

**Example :**

```
SQL>alter table student modify(sname varchar(10),class varchar(5));
```

**3. DROP :** This is used to delete the structure of a relation. It permanently deletes the records in the table.

**Syntax :**

```
Drop table relation_name;
```

**Example :**

```
SQL>drop table std;
```

**4. RENAME :** It is used to modify the name of the existing database object.

**Syntax :**

```
RENAME table old_relation_name TO new_relation_name;
```

**Example :**

```
SQL>rename table std to std1;
```

**5. TRUNCATE :** This command will remove the data permanently. But structure will not be removed.

**Syntax :**

```
truncate table <table_name>;
```

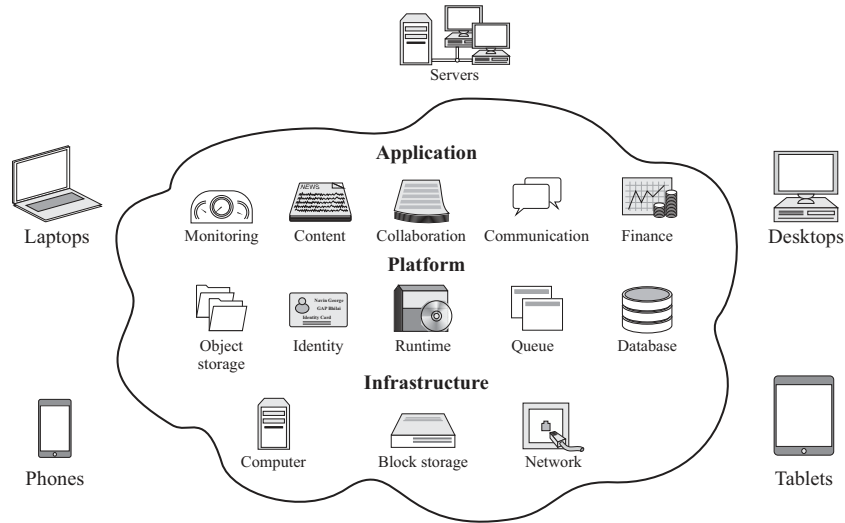
**Example :**

```
truncate table student;
```

**Q.7** (a) What is cloud computing? Explain the terms Cloud infrastructure.

**Ans. Cloud computing :** Cloud computing is a technology that uses the internet and central remote servers to maintain data and applications. Cloud computing allows consumers and businesses to use applications without installation and access their personal files at any computer with internet access. This technology allows more efficient computing by centralizing data storage, processing and bandwidth. Cloud computing customers do not own the physical infrastructure; rather they rent the usage from a third party provider. They consume resources as a service and pay only for resources that they use.

Some simple examples of cloud computing are **yahoomail**, **gmail** and **hotmail** etc. With an internet connection we can start sending e-mails. The server and e-mail management software is all on the cloud (internet) and is totally managed by the cloud service provider **yahoo**, **google** etc. The consumer get the software only and enjoy the benefits.



**Fig. : Cloud computing**

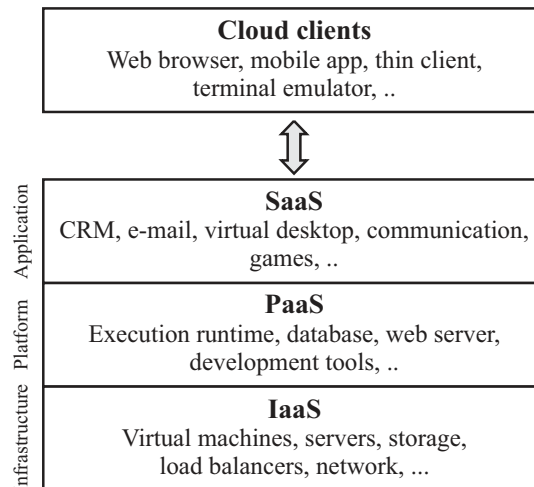
**Cloud infrastructure :** Cloud infrastructure refers to the hardware and software components such as **servers, storage, networking** and **virtualization software** that are needed to support the computing requirements of a cloud computing model. In addition, cloud infrastructure includes a software abstraction layer that virtualizes resources and logically presents them to users through programmatic means.

In cloud computing, virtualized resources are hosted by a service provider or IT department and delivered to users over a network or the internet. These resources include **virtual machines** and components such as **servers, computer, memory, network switches, firewalls, load balancers** and **storage**.

In a cloud computing architecture, cloud infrastructure consists of the back end components. Cloud infrastructure is present in each of the three main cloud computing models :

1. **Infrastructure as a service (IaaS),**
2. **Platform as a service (PaaS) and**
3. **Software as a service (SaaS).**

Together, these three models form what's often called a cloud computing stack, with IaaS as the foundation, PaaS as the middle layer and SaaS as the top layer.



**Fig. : Block diagram of cloud infrastructure**

Businesses use cloud infrastructures to run their applications. Unlike subscription based pricing models, or payment structures that enable users to subscribe to vendor services for a set price, cloud infrastructures are typically purchased using a pay-per-use model. In a pay-per-usage model, users only pay for the services consumed generally on an hourly, weekly or monthly basis. Businesses can also build cloud infrastructures on premises rather than purchase cloud infrastructure from a provider.

- Q.7** (b) Discuss the following terms :
- (i) Spamming
  - (ii) E-mail spoofing
  - (iii) Primary Key

**Ans. (i) Spamming :** Spamming is sending of unsolicited bulk commercial e-mail, junk news group postings, or instant message to a large number of people usually for advertising or marketing over the internet. The people who do this kind of activity are called **spammers**. They send thousands of e-mails to the receiver without their consent. The spammers can collect e-mail addresses by hacking or simply using computer programs that generate random addresses based on the domain names. Spamming is irritating but it is not illegal unless it causes damage to the system.

**(ii) E-mail spoofing :** E-mail spoofing is the forgery of an e-mail header so that the message appears to have originated from someone or somewhere other than the actual source. Distributors of spam often use spoofing in an attempt to get recipients to open, and possibly even respond to, their solicitations. Spoofing can be used legitimately. Spoofing anyone other than yourself is illegal in some jurisdictions.

E-mail spoofing is possible because simple mail transfer protocol (SMTP), the main protocol used in sending e-mail, does not include an authentication mechanism. Although an SMTP service extension (specified in IETF RFC 2554) allows an SMTP client to negotiate a security level with a mail server, this precaution is not often taken. If the precaution is not taken, anyone with the requisite knowledge can connect to the server and use it to send messages.

To send spoofed e-mail, senders insert commands in headers that will alter message information. It is possible to send a message that appears to be from anyone, anywhere, saying whatever the sender wants it to say. Thus, someone could send spoofed e-mail that appears to be from you with a message that you didn't write.

**(iii) Primary key :** A candidate key chosen by database designer as a principle means of identifying an entity in the relation is called as **primary key**. An entity type may have more than one candidate keys, in that case the database designer has to designate one of them as primary key, since there is always only a single primary key in an entity type. If there is just one candidate key then obviously the same will be declared as primary key.

**Example :** From the student table any one of the candidate key from {sno}, {rollno}, and {enrollno} is chosen to serve as primary key.

A certain value that may be associated with any attribute is NULL, that means "not given" or "not defined". A major characteristic of the PK is that it cannot have the NULL value. If PK is a composite, then none of the attributes included in the PK can have the NULL.

