

Government Polytechnic College, Jodhpur
Department of Computer Science (NBA Accredited)

Programme:Diploma

Class Test: III

Term: 2017-18

Course: Data Structure and Algorithm

Year: IIIrd

Course CODE: CS-301

Time: 7:45 to 8:45

Max.Marks :15

Date: 09-04-2018

Instructions to candidates: Attempt Any Three Questions

1.Mobiles,smart watches or any electronic gadgets are strictly banned.

Sl#	Question	Marks	CO MAPPING
1	What is Graph? Also explain BFS.	5	CO5
2	Explain Bubble Sorting Method.	5	CO4
3	Explain Prism Algorithm.	5	CO4
4.	Describe Binary Search Tree.	5	CO4

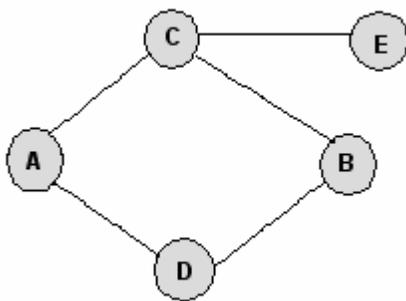
Ans 1:-

A graph G is a collection of two sets V and E where V is the collection of vertices v_0, v_1, \dots, v_{n-1} and E is the collection of edges e_1, e_2, \dots, e_n where an edge is an arc which connects two vertices. The vertices are also termed as nodes and edges are termed as arcs. This can be represented as

$$G = (V, E)$$

where, $V(G) = \{v_0, v_1, v_2, \dots, v_{n-1}\}$ and $E(G) = \{e_1, e_2, e_3, \dots, e_n\}$.

For example let us consider the graph $G = (V, E)$ shown in fig (12.1) where $V(G) = \{A, B, C, D, E\}$ and $E(G) = \{(A, C), (C, B), (B, D), (D, A), (C, E)\}$



Graph

Breadth First Search (BFS) algorithm traverses a graph in a breadthward motion and uses a queue to remember to get the next vertex to start a search, when a dead end occurs in any iteration.

- We use a queue Q to hold all gray vertices—vertices we have seen but are still not done with.
- We remember from which vertex a given vertex v is colored gray – i.e. the node that discovered v first; this is called parent[v].
- We also maintain d[v], the length of the path from s to v. Initially d[s] = 0.

BFS(s)

color[s] = gray

d[s] = 0

ENQUEUE(Q, s)

WHILE Q not empty DO

 DEQUEUE(Q, u)

 FOR each $v \in \text{adj}[u]$ DO

 IF color[v] = white THEN

 color[v] = gray

 d[v] = d[u] + 1

 parent[v] = u //(u,v) is a tree-edge

 ENQUEUE(Q, v)

 //ELSE v is not white, (u,v) is non-tree edge

 color[u] = black

Ans 2:- Bubble sort, sometimes referred to as sinking sort, is a simple sorting algorithm that repeatedly steps through the list to be sorted, compares each pair of adjacent items and swaps them if they are in the wrong order. The pass through the list is repeated until no swaps are needed, which indicates that the list is sorted. The algorithm, which is a comparison sort, is named for the way smaller or larger elements "bubble" to the top of the list. Although the algorithm is simple, it is too slow and impractical for most problems even when compared to sort. Bubble sort can be practical if the input is in mostly sorted order with some out-of-order elements nearly in position.

Algorithm for Bubble Sort: Bubble_Sort (A [] , N)

Step 1 : Repeat For P = 1 to N – 1 Begin

Step 2 : Repeat For J = 1 to N – P Begin

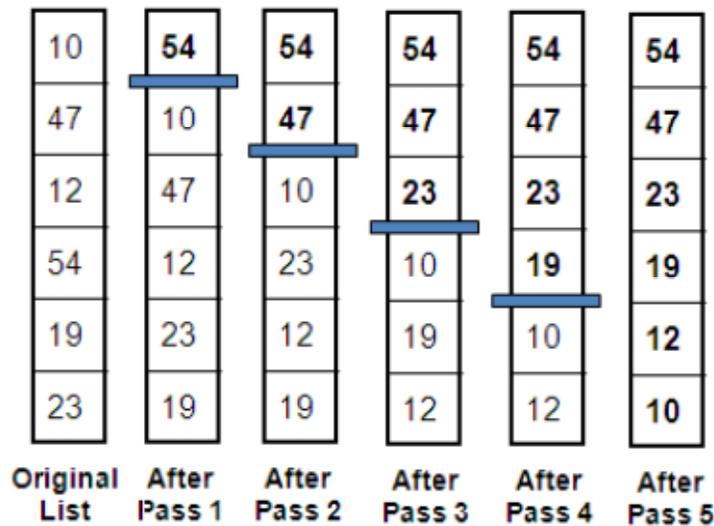
Step 3 : If (A [J] < A [J – 1])

Swap (A [J] , A [J – 1]) End For

End For

Step 4 : Exit

Ex:- A list of unsorted elements are: 10 47 12 54 19 23
(Bubble up for highest value shown here)



Ans 3:- The generic algorithm gives us an idea how to 'grow' a MST. If you read the theorem and the proof carefully, you will notice that the choice of a cut (and hence the corresponding light edge) in each iteration is immaterial. We can select any cut (that respects the selected edges) and find the light edge crossing that cut to proceed. The Prim's algorithm makes a nature choice of the cut in each iteration – it grows a single tree and adds a light edge in each iteration.

```

MST-PRIM(G, w, r)
1. for each u ∈ G.V
2.   u.key = ∞
3.   u.pi = NIL
4. r.key = 0
5. Q = G.V
6. while Q ≠ ∅
7.   u = EXTRACT-MIN(Q)
8.   for each v ∈ G.Adj[u]
9.     if v ∈ Q and w(u,v) < v.key
10.      v.pi = u
11.      v.key = w(u,v)

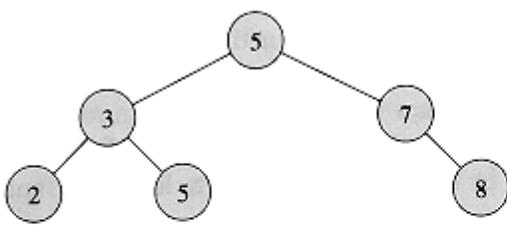
```

Basically the algorithm works as follows:

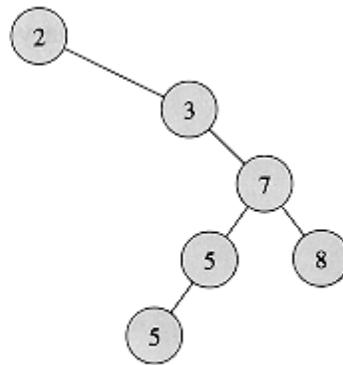
1. Initialize Q and set the source (root) key to 0
2. While Q is not empty, dequeue the vertex with minimum weight edge and add it to the tree by adding edge (u, π, u) to T
3. For each vertex v in Adj[u] that is still in Q, check if $w(u, v)$ (the edge weights from u for all vertices not in T) are less than the current v.key (the current smallest edge weight) and if so update the predecessor and key fields

The run time of Prim's algorithm depends on the implementation of the priority queue, but can be made to run in $O(E + V \lg V)$ using a *Fibonacci heap* for the priority queue.

Ans 4: A binary search tree is organized, as the name suggests, in a binary tree. Such a tree can be represented by a linked data structure in which each node is an object. In addition to a *key* field, each node contains fields *left*, *right*, and *p* that point to the nodes corresponding to its left child, its right child, and its parent, respectively. If a child or the parent is missing, the appropriate field contains the value NIL. The root node is the only node in the tree whose parent field is NIL.



(a)



(b)

Figure (a) Binary search trees. For any node x , the keys in the left subtree of x are at most $\text{key}[x]$, and the keys in the right subtree of x are at least $\text{key}[x]$. Different binary search trees can represent the same set of values. The worst-case running time for most search-tree operations is proportional to the height of the tree. (a) A binary search tree on 6 nodes with height 2. (b) A less efficient binary search tree with height 4 that contains the same keys.

The keys in a binary search tree are always stored in such a way as to satisfy the **binary-search-tree property**:

Let x be a node in a binary search tree. If y is a node in the left subtree of x , then $\text{key}[y] \leq \text{key}[x]$. If y is a node in the right subtree of x , then $\text{key}[x] \leq \text{key}[y]$.

Thus, in Figure (a), the key of the root is 5, the keys 2, 3, and 5 in its left subtree are no larger than 5, and the keys 7 and 8 in its right subtree are no smaller than 5. The same property holds for every node in the tree. For example, the key 3 in Figure (a) is no smaller than the key 2 in its left subtree and no larger than the key 5 in its right subtree.

The binary-search-tree property allows us to print out all the keys in a binary search tree in sorted order by a simple recursive algorithm, called an **inorder tree walk**. This algorithm derives its name from the fact that the key of the root of a subtree is printed between the values in its left subtree and those in its right subtree. (Similarly, a **preorder tree walk** prints the root before the values in either subtree, and a **postorder tree walk** prints the root after the values in its subtrees.) To use the following procedure to print all the elements in a binary search tree T , we call `INORDER-TREE-WALK(root[T])`.

Programme: Diploma

Class Test: III

Term: 2017-18

Course: Object Oriented Programming through C++

Year: IIIrd

Course CODE: CS-302

Time: 09:00 to 10:00

Max.Marks : 15

Date: 09-04-2018

Instructions to candidates: Attempt Any Three Questions

1. Mobiles, smart watches or any electronic gadgets are strictly banned.

Sl#	Question	Marks	CO MAPPING
1	Define Inheritance. Explain Types of inheritance.	5	CO5
2	What is Virtual function? Explain with programming example.	5	CO5
3	Explain Exception handling mechanism with programming example.	5	CO6
4.	Define C++ streams. What are the stream classes available in C++?	5	CO6

1. Define Inheritance. Explain Types of inheritance.

Inheritance - Inheritance in Object Oriented Programming can be described as a process of creating new classes from existing classes. New classes inherit some of the properties and behavior of the existing classes

Type of Inheritance -

1. Single Inheritance –

<p>In this inheritance, a derived class is created from a single base class.</p> <div style="text-align: center;"> <pre> graph TD BC[Base Class] --> DC[Derived Class] </pre> </div>	<pre> class base { }; class derived: public base { }; </pre>
---	--

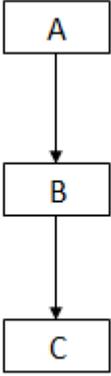
2. Multiple inheritance –

<p>In this inheritance, a derived class is created from more than one base class</p> <div style="text-align: center;"> <pre> graph TD B1[Base1] --> D[Derived] B2[Base2] --> D </pre> </div>	<pre> class base1 { }; class base2 { }; class derived: public base1, public base2 { }; </pre>
--	---

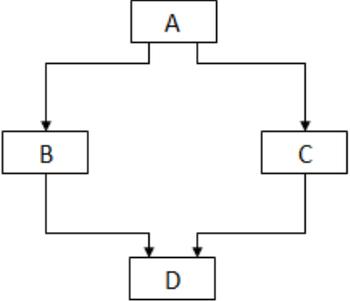
3. Hierarchical

<p>In this inheritance, more than one derived classes are created from a single base.</p> <div style="text-align: center;"> <pre> graph TD A[A] --> B[B] A --> C[C] A --> D[D] </pre> </div>	<pre> class base { }; class derived1: public base { }; class derived2: public base { }; </pre>
---	--

4. Multilevel -

<p>In this inheritance, a derived class is created from another derived classes and the same base class of another derived classes.</p>  <p>Multilevel Inheritance</p>	<pre>class base { }; class derived1: public base { }; class derived2: public derived1 { };</pre>
--	--

5. Hybrid –

<p><i>In this inheritance, more than one derived classes are created from a single base class.</i></p>  <p>Hybrid Inheritance</p>	<pre>class base { }; class derived1: virtual public base { }; class derived2: virtual public base { }; class final: public derived1,public derived2 { };</pre>
---	--

2. What is Virtual function? Explain with programming example.

A virtual function a member function which is declared within base class and is re-defined (Overriden) by derived class. When we refer to a derived class object using a pointer or a reference to the base class, we can call a virtual function for that object and execute the derived class's version of the function. Virtual functions ensure that the correct function is called for an object, regardless of the type of reference (or pointer) used for function call. They are mainly used to achieve Runtime polymorphism Functions are declared with a **virtual** keyword in base class. The resolving of function call is done at Run-time.

```
class base
{
public:
    virtual void print ()
    { cout<< "print base class" <<endl; }
};
class derived:public base
{
public:
    void print ()
    { cout<< "print derived class" <<endl; }
```

```

};
int main()
{
    base *bptr,b;
    derived d;
    bptr = &b;
    //virtual function, binded at runtime
    bptr->print();

    bptr = &d;
    //virtual function, binded at runtime
    bptr->print();
}

```

Runtime polymorphism is achieved only through a pointer (or reference) of base class type. Also, a base class pointer can point to the objects of base class as well as to the objects of derived class. In above code, base class pointer 'bptr' contains the address of object 'd' of derived class. If we have created virtual function in base class and it is being overridden in derived class then we don't need virtual keyword in derived class, functions are automatically considered as virtual functions in derived class.

3. Explain Exception handling mechanism with programming example.

A C++ program is comprised of many types of error. Runtime errors occur while executing a program. Main purpose of the Exception handling mechanism is to provide means to detect and report an "exceptional circumstance" so that an appropriate action can be taken.

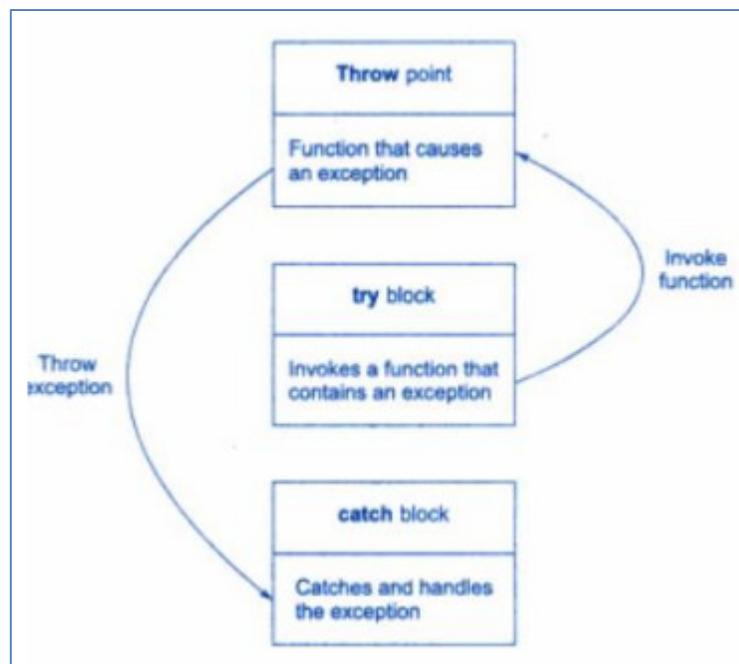
Exceptions are some peculiar problems that may occur other than basic syntax and logical errors. For handling these exceptions there are built in language features provided by ANSI C++. Original C++ does not have the features of exception handling but this feature is present in ANSI C++.

Basically there are three keywords in error handling code they are "try, throw and catch". 'Try' keyword is used to preface a block of statements that generates exception. The statements are surrounded by braces. 'catch' catches and handles the exceptions that are thrown by 'throw' from try block as shown in the figure below. After the try block throws an exception the program enters the catch statement. Syntax of the exception mechanism is as shown in the figure below.

```

try
{
    .....
    throw exception
    //Block of statements which detects and throws an exception
    .....
    .....
}
catch(type arg)
{
    .....
    //Block of statements that handles the exception
    .....
}

```



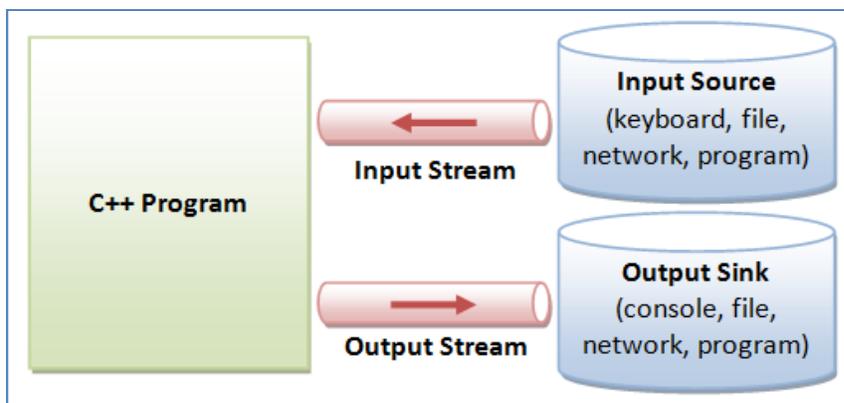
```

#include<iostream.h>
int main()
{
  int x,y,z;
  cout<<"Enter the values of x and y:";
  cin>>x>>y;
  z=x-y;
  try
  {
    if(z!=0)
    {
      cout<<"Result(x/z)="<<x/z;
    }
    else          //division by zero exception
    {
      throw(z);    //throws int object
    }
  }

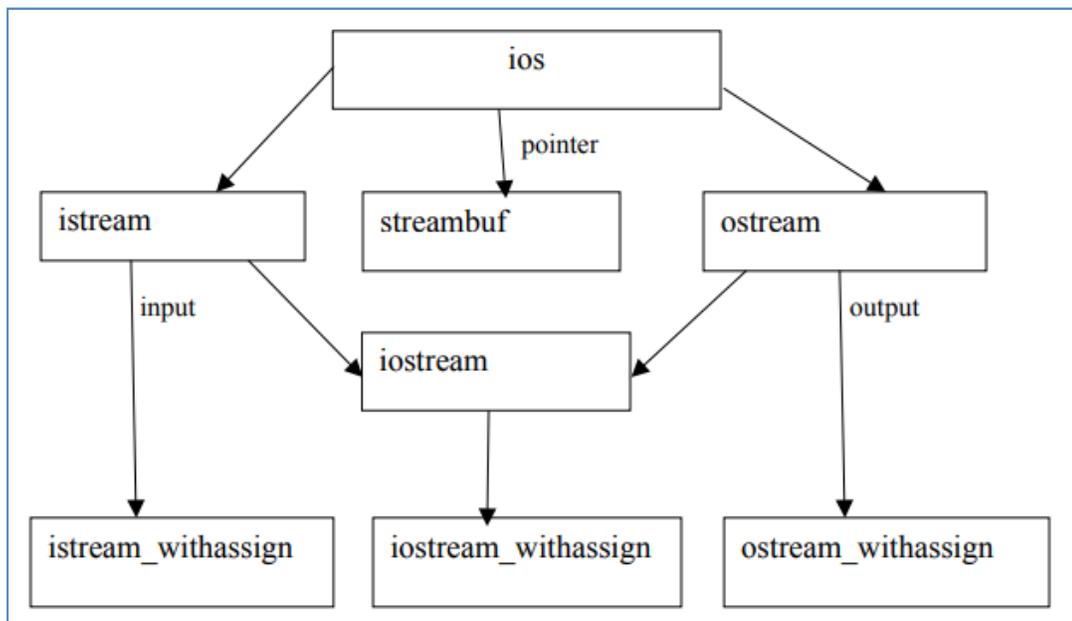
  catch(int)      //catches the exception
  {
    cout<<"Exception caught: z="<<z;
  }
  return 0;
}
  
```

4. Define C++ streams. What are the stream classes available in C++?

C/C++ IO are based on *streams*, which are sequence of bytes flowing in and out of the programs (just like water and oil flowing through a pipe). In input operations, data bytes flow from an *input source* (such as keyboard, file, network or another program) into the program. In output operations, data bytes flow from the program to an *output sink* (such as console, file, network or another program). Streams acts as an intermediaries between the programs and the actual IO devices, in such the way that frees the programmers from handling the actual devices, so as to archive device independent IO operations.



The C++ I/O system contains a hierarchy of classes that are used to define various streams to deal with both the console and disk files. These classes are called stream classes. Figure shows the hierarchy of the stream classes used for input and output operations with the console unit. These classes are declared in the header file `iostream`. The file should be included in all programs that communicate with the console unit.



As in figure `ios` is the base class for `istream`(input stream) and `ostream`(output stream) which are base classes for `iostream`(input/output stream). The class `ios` is declared as the virtual base class so that only one copy of its members are inherited by the `iostream`. The class `ios` provides the basic support for formatted and unformatted input/output operations. The class `istream` provides the facilities for formatted and unformatted input while the class `ostream`(through inheritance) provides the facilities for formatted output. The class `iostream` provides the facilities for handling both input output streams. Three classes namely `istream_withassign`, `ostream_withassign` and `iostream_withassign` add assignment operators to these classes.

304: SOFTWARE ENGINEERING THIRD TEST SOLUTION

Government Polytechnic College, Jodhpur
Department of Computer Science (NBA Accredited)

Programme: Diploma
2017-18

Class Test: III

Session:

Course: Software Engineering

Year: IIIrd

Course CODE: CS-304

Time:

09:00 to 10:00

Max.Marks : 15

Date: 10-

04-2018

Instructions to candidates: Attempt Any Three Questions

Sl#	Question	Marks	CO MAPPING
1	What is ISO9000 for Reliability? Write requirements of it.	5	CO5
2	What are Cohesion and Coupling? Explain different Cohesion Techniques.	5	CO3
3	Explain Six Sigma.	5	CO4
4.	Describe McCabe's Cyclomatic Complexity Metric.	5	CO5

Ans. 1

ISO 9000 is a series of three standards: ISO 9001, ISO 9002, and ISO 9003. The ISO 9000 series of standards is based on the premise that if a proper process is followed for production, then good quality products are bound to follow automatically. The types of industries to which the different ISO standards apply are as follows. ISO 9001 applies to the organizations engaged in design, development, production, and servicing of goods. This is the standard that is applicable to most software development organizations. ISO 9002 applies to those organizations which do not design products but are only involved in production. Examples of these category industries include steel and car manufacturing industries that buy the product and plant designs from external sources and are involved in only manufacturing those products. Therefore, ISO 9002 is not applicable to software development organizations. ISO 9003 applies to organizations that are involved only in installation and testing of the products.

Ans. 2

COUPLING

An indication of the strength of interconnections between program units.

Highly coupled have program units dependent on each other. Loosely coupled are made up of units that are independent or almost independent.

Modules are independent if they can function completely without the presence of the other. Obviously, can't have modules completely independent of each other. Must interact so that can produce desired outputs. The more connections between

modules, the more dependent they are in the sense that more info about one modules is required to understand the other module.

Three factors: number of interfaces, complexity of interfaces, type of info flow along interfaces.

Want to minimize number of interfaces between modules, minimize the complexity of each interface, and control the type of info flow. An interface of a module is used to pass information to and from other modules.

In general, modules tightly coupled if they use shared variables or if they exchange control info.

Loose coupling if info held within a unit and interface with other units via parameter lists. Tight coupling if shared global data.

If need only one field of a record, don't pass entire record. Keep interface as simple and small as possible.

COHESION

Measure of how well module fits together.

A component should implement a single logical function or single logical entity. All the parts should contribute to the implementation.

Many levels of cohesion:

1. **Coincidental cohesion:** the parts of a component are not related but simply bundled into a single component.

harder to understand and not reusable.

2. **Logical association:** similar functions such as input, error handling, etc. put together. Functions fall in same logical class. May pass a flag to determine which ones executed.

interface difficult to understand. Code for more than one function may be intertwined, leading to severe maintenance problems. Difficult to reuse

3. **Temporal cohesion:** all of statements activated at a single time, such as start up or shut down, are brought together. Initialization, clean up.

Functions weakly related to one another, but more strongly related to functions in other modules so may need to change lots of modules when do maintenance.

4. Procedural cohesion: a single control sequence, e.g., a loop or sequence of decision statements. Often cuts across functional lines. May contain only part of a complete function or parts of several functions.

Functions still weakly connected, and again unlikely to be reusable in another product.

5. Communicational cohesion: operate on same input data or produce same output data. May be performing more than one function. Generally acceptable if alternate structures with higher cohesion cannot be easily identified.

still problems with reusability.

6. Sequential cohesion: output from one part serves as input for another part. May contain several functions or parts of different functions.
7. Informational cohesion: performs a number of functions, each with its own entry point, with independent code for each function, all performed on same data structure. Different than logical cohesion because functions not intertwined.
8. Functional cohesion: each part necessary for execution of a single function. e.g., compute square root or sort the array.

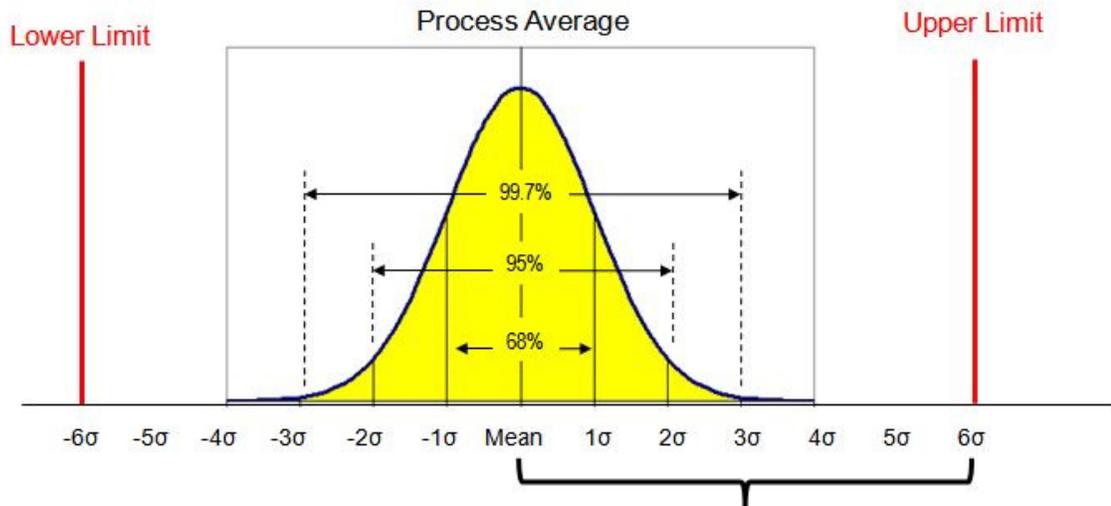
Usually reusable in other contexts. Maintenance easier.

9. Type cohesion: modules that support a data abstraction.

Not strictly a linear scale. Functional much stronger than rest while first two much weaker than others. Often many levels may be applicable when considering two elements of a module. Cohesion of module considered as highest level of cohesion that is applicable to all elements in the module.

Ans. 3

Six Sigma is a disciplined, statistical-based, data-driven approach and continuous improvement methodology for eliminating defects in a product, process or service. It was developed by Motorola in early to middle 1980's based on quality management fundamentals, then became a popular management approach at General Electric (GE) in the early 1990's. Hundreds of companies around the world have adopted Six Sigma as a way of doing business. Sigma represents the population standard deviation, which is a measure of the variation in a data set collected about the process. If a defect is defined by specification limits separating good from bad outcomes of a process, then a six sigma process has a process mean (average) that is six standard deviations from the nearest specification limit. This provides enough buffer between the process natural variation and the specification limits.



For example, if a product must have a thickness between 10.32 and 10.38 inches to meet customer requirements, then the process mean should be around 10.35, with a standard deviation less than 0.005 (10.38 would be 6 standard deviations away from 10.35).

Six Sigma can also be thought of as a measure of process performance, with Six Sigma being the goal, based on the defects per million. Once the current performance of the process is measured, the goal is to continually improve the sigma level striving towards 6 sigma. Even if the improvements do not reach 6 sigma, the improvements made from 3 sigma to 4 sigma to 5 sigma will still reduce costs and increase customer satisfaction.

Sigma Level	Defects per Million	Yield
6	3.4	99.99966%
5	230	99.977%
4	6,210	99.38%
3	66,800	93.32%
2	308,000	69.15%
1	690,000	30.85%

Ans. 4

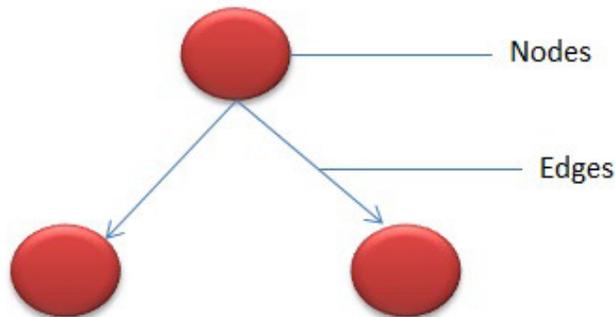
Cyclomatic complexity is a software metric used to measure the complexity of a program. This metric measures independent paths through program source code.

Independent path is defined as a path that has at least one edge which has not been traversed before in any other paths.

Cyclomatic complexity can be calculated with respect to functions, modules, methods or classes within a program.

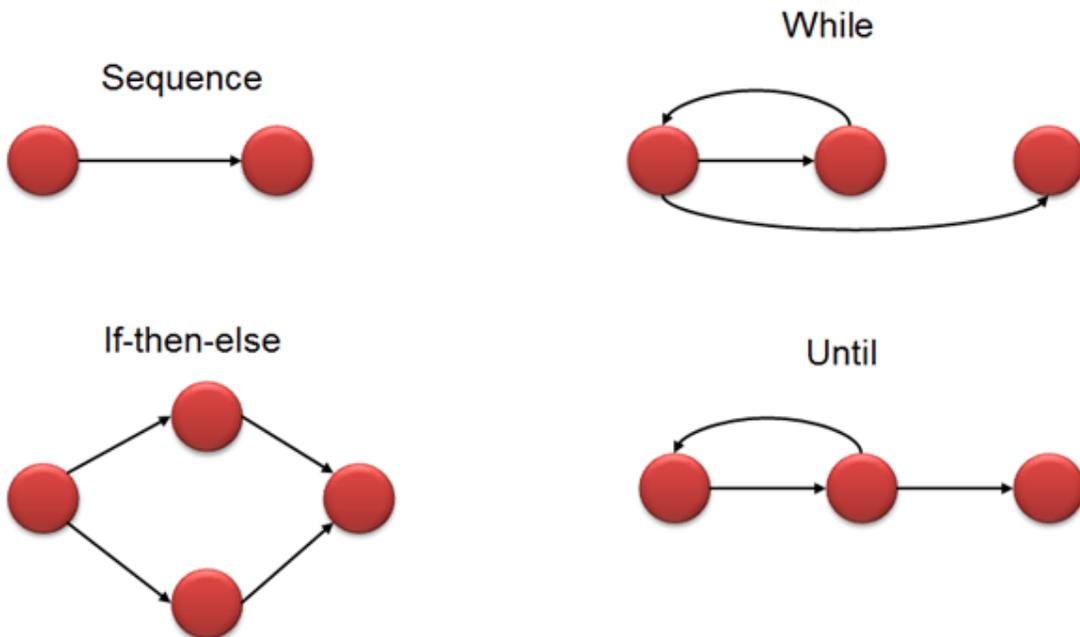
This metric was developed by Thomas J. McCabe in 1976 and it is based on a control flow representation of the program. Control flow depicts a program as a graph which consists of Nodes and Edges.

In the graph, Nodes represent processing tasks while edges represent control flow between the nodes.



Flow graph notation for a program:

Flow Graph notation for a program is defines. several nodes connected through the edges. Below are Flow diagrams for statements like if-else, While, until and normal sequence of flow.



Mathematical representation:

Mathematically, it is set of independent paths through the graph diagram. The complexity of the program can be defined as -

$$V(G) = E - N + 2$$

Where,

E - Number of edges

N - Number of Nodes

$$V(G) = P + 1$$

Where P = Number of predicate nodes (node that contains condition)

Government Polytechnic College, Jodhpur
Department of Computer Science (NBA Accredited)

Programme: Diploma
 2017-2018

Class Test: III

Session:

Course: DOT NET TECHNOLOGY

Year: IIIrd

Course CODE: CS-305

Time: 7:45 to 8:45

Max.Marks : 15

Date: 04-04-2018

Instructions to candidates: Attempt any three Questions

1. Mobiles, smart watches or any electronic gadgets are strictly banned in examination hall.

SI#	Question	Marks	CO MAPPING
1	Briefly explain the properties and methods of Timer control in VB.NET with example.	5	CO5
2	What is AdO.NET? Compare ADO with ADO.NET.	5	CO5
3	What are similarities and differences between ASP and ASP.NET.	5	CO6
4	Explain architecture of ASP.NET in detail.	5	CO6

Solution

Q1. Briefly explain the properties and methods of Timer control in VB.NET with example.

Sol. Timer Control

A Timer control is invisible at run time, and its purpose is to send a periodic pulse to the current application. You can trap this pulse by writing code in the Timer's Timer event procedure and take advantage of it to execute a task in the background or to monitor a user's actions. This control exposes only two meaningful properties: Interval and Enabled. Interval stands for the number of milliseconds between subsequent pulses (Timer events), while Enabled lets you activate or deactivate events. When you place the Timer control on a form, its Interval is 0, which means no events. Therefore, remember to set this property to a suitable value in the Properties window or in the Form_Load event procedure:

Timer Control Properties

Properties:

Properties	Description
Enabled	Property used to Get or set whether the timer is running.
Interval	Property used to set or get the time in millisecond between the timer clicks.

Methods:

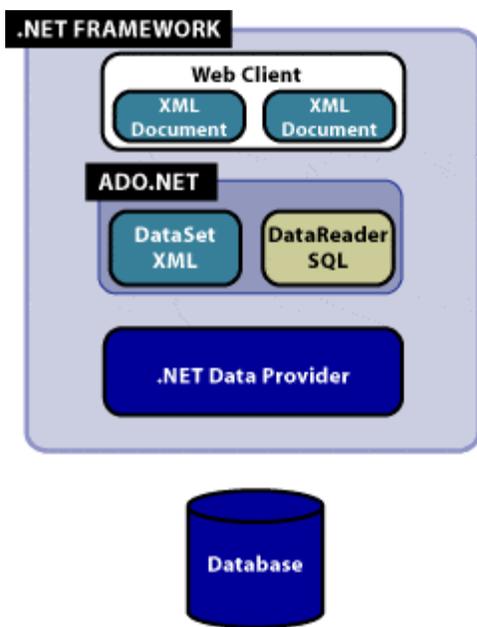
Events	Description
Start	Method used to start timer.
Stop	Method used to stop timer.

Events:

Events	Description
Tick	Triggered when the time interval has elapsed.

Q2. What is AdO.NET? Compare ADO with ADO.NET..

Sol. ADO.NET is made of a set of classes that are used for connecting to a database, providing access to relational data, XML, and application data, and retrieving results. ADO.NET data providers contain classes that represent the provider's Connection, Command, DataAdapter and DataReader objects (among others). The following figure shows how ADO.NET fits into the communication between a Web client and the DBMS.



ADO	ADO.Net
ADO is base on COM : Component Object Modelling based.	ADO.Net is based on CLR : Common Language Runtime based.
ADO stores data in binary format.	ADO.Net stores data in XML format i.e. parsing of data.
ADO can't be integrated with XML because ADO have limited access of XML.	ADO.Net can be integrated with XML as having robust support of XML.
In ADO, data is provided by RecordSet .	In ADO.Net data is provided by DataSet or DataAdapter .
ADO is connection oriented means it requires continuous active connection.	ADO.Net is disconnected , does not need continuous connection.
ADO gives rows as single table view, it scans sequentially the rows using MoveNext method.	ADO.Net gives rows as collections so you can access any record and also can go through a table via loop.
In ADO, You can create only Client side cursor.	In ADO.Net, You can create both Client & Server side cursor.
Using a single connection instance, ADO can not handle multiple transactions.	Using a single connection instance, ADO.Net can handle multiple transactions.

Q3. What are similarities and differences between ASP and ASP.NET.

Sol. ASP or classic ASP was a Microsoft technology in that all logic was embedded in HTML pages using ASP tags. Logic mostly could be written using VBScript and sometimes JScript (MS dialect of JavaScript). It was counterpart to earlier releases of JSP (Java Server Pages). Since it was based on Component Object Model (COM) many libraries (from Microsoft) could be used as well. You can identify a page that uses ASP when the page has .asp extension.

ASP.NET is totally different technology from Microsoft based on .NET platform. .NET platform consists of .NET framework (languages like C#, VB.NET, F#, Base Class Library, Common Language Runtime,...) and other MS based ecosystem (IIS as a web server, SQL server as database, Windows as operating system,...).

ASP:

- i. It has limited oops support and not having built in support for xml.
- ii. Very less development and debugging tool available. Difficult to debug the code.
- iii. ASP you can only do scripting using visual basic scripting and java scripting.
- iv. Error handling is very poor.
- v. It has no high level programming structure. Mixed of html and server side scripting.
- vi. It has no in built validation control. Validating page is difficult for developers.
- vii. In the classic ASP if you need to update code on the existing page then it is mandatory to restart the server to get reflect.

ASP.NET

- i. ASP.NET is full featured object oriented programming.
- ii. It has full support of xml. Which helps easy data exchange.
- iii. Various tools and compiler available. Microsoft Visual studio makes your debugging job easier.
- iv. ASP.NET we can use either C# or VB.NET as server side programming language.
- v. ASP.NET gives you three tire architecture. It allow you to keep your business logic, views everything separate. Meaning that easy to enhance applications.
- vi. Error handling is very good.
- vii. It has state management support.

viii. In built validation controls. It has rich validation set - custom validator, range validator, regular expression, compare and require field validation control which makes your job easier.

Q4. Explain architecture of ASP.NET in detail..

Sol. ASP.NET works on three tier architecture. This architecture is extremely popular because they increase application performance, scalability, flexibility, and code reuse. In three tier architecture, applications are divided into three major areas of functionality:

- The data layer manages the data storage and retrieval.
- The business layer maintains business rules and logic
- The presentation layer manages the user interface and related presentation code.

The presentation Layer

Presentation layer provides the interface to the users of website or application. It contains pages like .aspx or Windows forms where data is presented to the user or user enters the information. This layer communicates with business layer. In ASP.NET the code behind file and HTML file are separate file. The HTML file defines the look and layout of the web form and the code behind file contains the presentation logic. This separation helps designers and developers. Designers don't have to worry about business logic to make user interface changes, and developers don't have to worry about to update code.

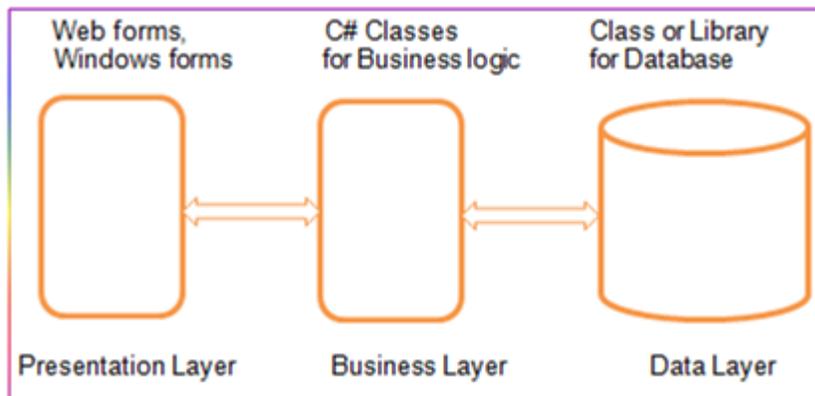
The business Layer

Business layer or Application layer acts between Application layer and Data Access Layer. This layer contains our logic, validating the data and other functions. In this layer programmer access the data access layer (also known as the DAL), functionality. As example suppose any user wants to register and he fills all the detail and click the submit button on presentation layer then call goes to business layer. In business layer programmers only access the function and pass the parameter that is provided by the end user as example.

InserUserDetails(string Username, string Password, string Email, string Firstname, string Lastname, string Address, float Salary);

The data Layer

All the code related to database is written in this layer. It contains methods to connect with database. It also contains the methods to perform insert, update, delete, select data according to the SQL query.



The main advantages of the 3 Tier Architecture is as follows:

Flexibility – All the layers have separate code so it makes the application much more flexible to changes.

Maintainability – You can change to the components in one layer and it will no effect on any others layers. So you can maintain the application very easily.

Reusability – A single library/component may be accessed by multiple components in the presentation layer. It easier to implement re-usable components by separating the application into multiple layers makes.

Scalability – You can deploy application components across multiple servers, so it makes the system much more scalable.

306: COMPUTER NETWORK THIRD TEST SOLUTION

Government Polytechnic College, Jodhpur

Department of Computer Science (NBA Accredited)

Programme: Diploma

Class Test: III

Session: 2017-18

Course: Computer Network

Year: IIIrd

Course CODE: CS-306

Time: 09:00 to 10:00

Max.Marks : 15

Date: 11-04-2018

Instructions to candidates: Attempt Any Three Questions

SI#	Question	Marks	CO MAPPING
1	Explain 802.11 standard.	5	CO5
2	Explain TCP header format.	5	CO3
3	Explain different transition diagrams of Transport Layer Reliable Delivery.	5	CO3
4.	Write short notes on Bluetooth and WiMAX.	5	CO5

Ans. 1

IEEE 802.11 is a set of media access control (MAC) and physical layer (PHY) specifications for implementing wireless local area network (WLAN) computer communication in the 900 MHz and 2.4, 3.6, 5, and 60 GHz frequency bands. They are the world's most widely used wireless computer networking standards, used in most home and office networks to allow laptops, printers, and smartphones to talk to each other and access the Internet without connecting wires. They are created and maintained by the Institute of Electrical and Electronics Engineers (IEEE) LAN/MAN Standards Committee (IEEE 802). The base version of the standard was released in 1997, and has had subsequent amendments. The standard and amendments provide the basis for wireless network products using the Wi-Fi brand. While each amendment is officially revoked when it is incorporated in the latest version of the standard, the corporate world tends to market to the revisions because they concisely denote capabilities of their products. As a result, in the marketplace, each revision tends to become its own standard.

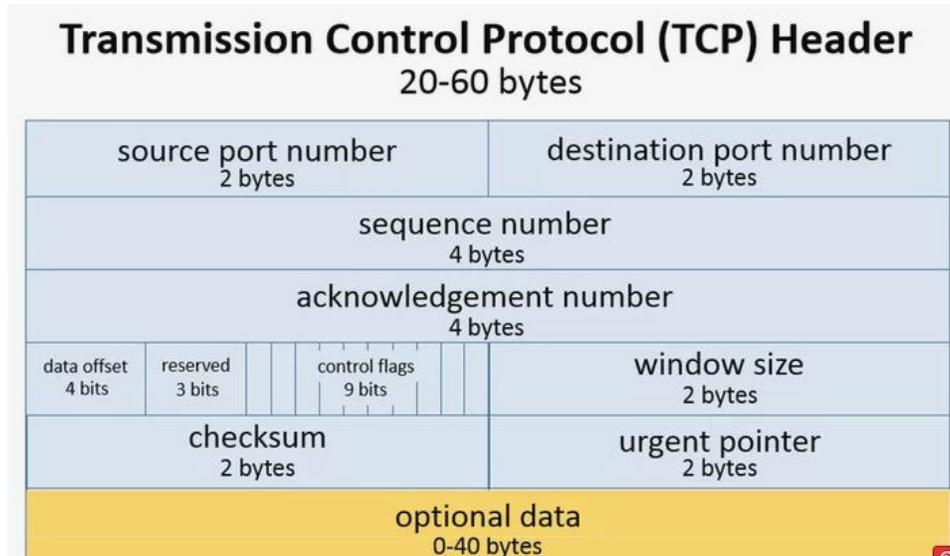
The 802.11 family consists of a series of half-duplex over-the-air modulation techniques that use the same basic protocol. 802.11-1997 was the first wireless networking standard in the family, but 802.11b was the first widely accepted one, followed by 802.11a, 802.11g, 802.11n, and 802.11ac. Other standards in the family (c-f, h, j) are service amendments that are used to extend the current scope of the existing standard, which may also include corrections to a previous specification.

802.11b and 802.11g use the 2.4 GHz ISM band, operating in the United States under Part 15 of the U.S. Federal Communications Commission Rules and Regulations. Because of this choice of frequency band, 802.11b and g equipment may occasionally suffer interference from microwave ovens, cordless telephones, and Bluetooth devices. 802.11b and 802.11g control their interference and susceptibility to interference by using direct-sequence spread spectrum (DSSS) and orthogonal frequency-division multiplexing (OFDM) signaling methods, respectively. 802.11a uses the 5 GHz U-NII band, which, for much of the world, offers at least 23 non-overlapping channels rather than the 2.4 GHz ISM frequency band offering only three non-overlapping channels, where other adjacent channels overlap—see list of WLAN channels. Better or worse performance with higher or lower frequencies (channels) may be realized, depending on the environment. 802.11n can use either the 2.4 GHz or the 5 GHz band; 802.11ac uses only the 5 GHz band.

The segment of the radio frequency spectrum used by 802.11 varies between countries. In the US, 802.11a and 802.11g devices may be operated without a license, as allowed in Part 15 of the FCC Rules and Regulations. Frequencies used by channels one through six of 802.11b and 802.11g fall within the 2.4 GHz amateur radio band. Licensed amateur radio operators may operate 802.11b/g devices under Part 97 of the FCC Rules and Regulations, allowing increased power output but not commercial content or encryption.

Ans. 2

TCP Header Format



Each TCP header has ten required fields totaling 20 bytes (160 bits) in size.

They can also optionally include an additional data section up to 40 bytes in size.

This is the layout of TCP headers:

1. Source TCP port number (2 bytes)
2. Destination TCP port number (2 bytes)
3. Sequence number (4 bytes)
4. Acknowledgment number (4 bytes)
5. TCP data offset (4 bits)
6. Reserved data (3 bits)
7. Control flags (up to 9 bits)
8. Window size (2 bytes)
9. TCP checksum (2 bytes)
10. Urgent pointer (2 bytes)
11. TCP optional data (0-40 bytes)

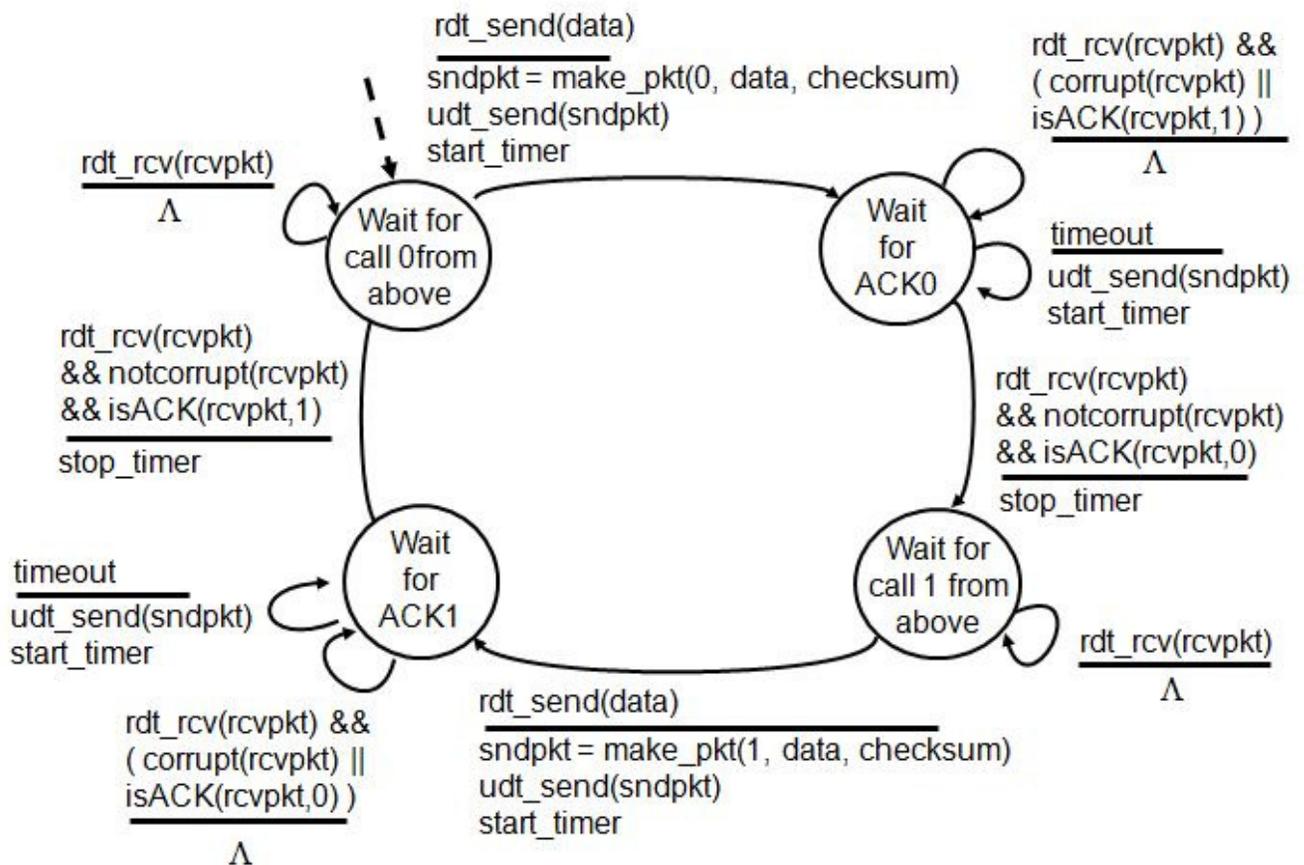
TCP inserts header fields into the message stream in the order listed above.

- *Source and destination TCP port numbers* are the communication endpoints for sending and receiving devices.
- Message senders use *sequence numbers* to mark the ordering of a group of messages. Both senders and receivers use the acknowledgment *numbers* field to communicate the sequence numbers of messages that are either recently received or expected to be sent.
- The *data offset field* stores the total size of a TCP header in multiples of four bytes. A header not using the optional TCP field has a data offset of 5 (representing 20 bytes), while a header using the maximum-sized optional field has a data offset of 15 (representing 60 bytes).
- *Reserved data* in TCP headers always has a value of zero. This field serves the purpose of aligning the total header size as a multiple of four bytes (important for the efficiency of computer data processing).
- TCP uses a set of six standard and three extended *control flags* (each an individual bit representing *on* or *off*) to manage data flow in specific situations. One bit flag, for example, initiates TCP connection reset logic. The detailed operation of these fields goes beyond the scope of this article.
- TCP senders use a number called *window size* to regulate how much data they send to a receiver before requiring an acknowledgment in return. If the window size becomes too small, network data transfer will be unnecessarily slow, while if the window size becomes too large, the network link can become saturated (unusable for any other

applications) or the receiver may not be able to process incoming data quickly enough (also resulting in slow performance). Windowing algorithms built into the protocol dynamically calculate size values and use this field of TCP headers to coordinate changes between senders and receivers.

- The *checksum* value inside a TCP header is generated by the protocol sender as a mathematical technique to help the receiver detect messages that are corrupted or tampered with.
- The urgent pointer field is often set to zero and ignored, but in conjunction with one of the control flags, it can be used as a data offset to mark a subset of a message as requiring priority processing.
- Usages of optional TCP data go beyond the scope of this article but include support for special acknowledgment and window scaling algorithms.

Ans. 3



Ans. 4

WiMAX base station equipment with a sector antenna and wireless modem on top

WiMAX (Worldwide Interoperability for Microwave Access) is a family of wireless communication standards based on the IEEE 802.16 set of standards, which provide multiple physical layer (PHY) and Media Access Control (MAC) options.

The name "WiMAX" was created by the **WiMAX Forum**, which was formed in June 2001 to promote conformity and interoperability of the standard, including the definition of predefined system profiles for commercial vendors. The forum describes WiMAX as "a standards-based technology enabling the delivery of last mile wireless broadband access as an alternative to cable and DSL". IEEE 802.16m or WirelessMAN-Advanced was a candidate for the 4G, in competition with the LTE Advanced standard.

WiMAX was initially designed to provide 30 to 40 megabit-per-second data rates, with the 2011 update providing up to 1 Gbit/s for fixed stations.

WiMAX refers to interoperable implementations of the IEEE 802.16 family of wireless-networks standards ratified by the WiMAX Forum. (Similarly, Wi-Fi refers to interoperable implementations of the IEEE 802.11 Wireless LAN standards certified by the Wi-Fi Alliance.) WiMAX Forum certification allows vendors to sell fixed or mobile products as WiMAX

certified, thus ensuring a level of interoperability with other certified products, as long as they fit the same profile.

The original IEEE 802.16 standard (now called "Fixed WiMAX") was published in 2001. WiMAX adopted some of its technology from WiBro, a service marketed in Korea.

Mobile WiMAX (originally based on 802.16e-2005) is the revision that was deployed in many countries, and is the basis for future revisions such as 802.16m-2011.

WiMAX is sometimes referred to as "Wi-Fi on steroids" and can be used for a number of applications including broadband connections, cellular backhaul, hotspots, etc. It is similar to Long-range Wi-Fi, but it can enable usage at much greater distances.

Bluetooth: **Bluetooth** is a wireless technology standard for exchanging data over short distances (using short-wavelength UHF radio waves in the ISM band from 2.4 to 2.485 GHz) from fixed and mobile devices, and building personal area networks (PANs). Invented by telecom vendor Ericsson in 1994, it was originally conceived as a wireless alternative to RS-232 data cables.

Bluetooth is managed by the Bluetooth Special Interest Group (SIG), which has more than 30,000 member companies in the areas of telecommunication, computing, networking, and consumer electronics. The IEEE standardized Bluetooth as **IEEE 802.15.1**, but no longer maintains the standard. The Bluetooth SIG oversees development of the specification, manages the qualification program, and protects the trademarks. A manufacturer must meet Bluetooth SIG standards to market it as a Bluetooth device. A network of patents apply to the technology, which are licensed to individual qualifying devices.

Government Polytechnic College, Jodhpur
Department of Computer Science (NBA Accredited)

Programme: Diploma
 2017-18

Class Test: III

Term:

Course: DATA WAREHOUSE AND MINING

Year: IIIrd

Course CODE: CS-307

Time: 7:45 to 8:45

Max.Marks :15

Date:05-04-2017

Instructions to candidates: Attempt any three Questions

1. Mobiles, smart watches or any electronic gadgets are strictly banned in examination hall.

SI#	Question	Marks	CO MAPPING
1	Explain the three-tier data warehouse architecture.	5	CO5
2	Explain various access tools in brief.	5	CO5
3	Differentiate between OLAP and OLTP.	5	CO6
4	Explain ROLAP server.	5	CO6

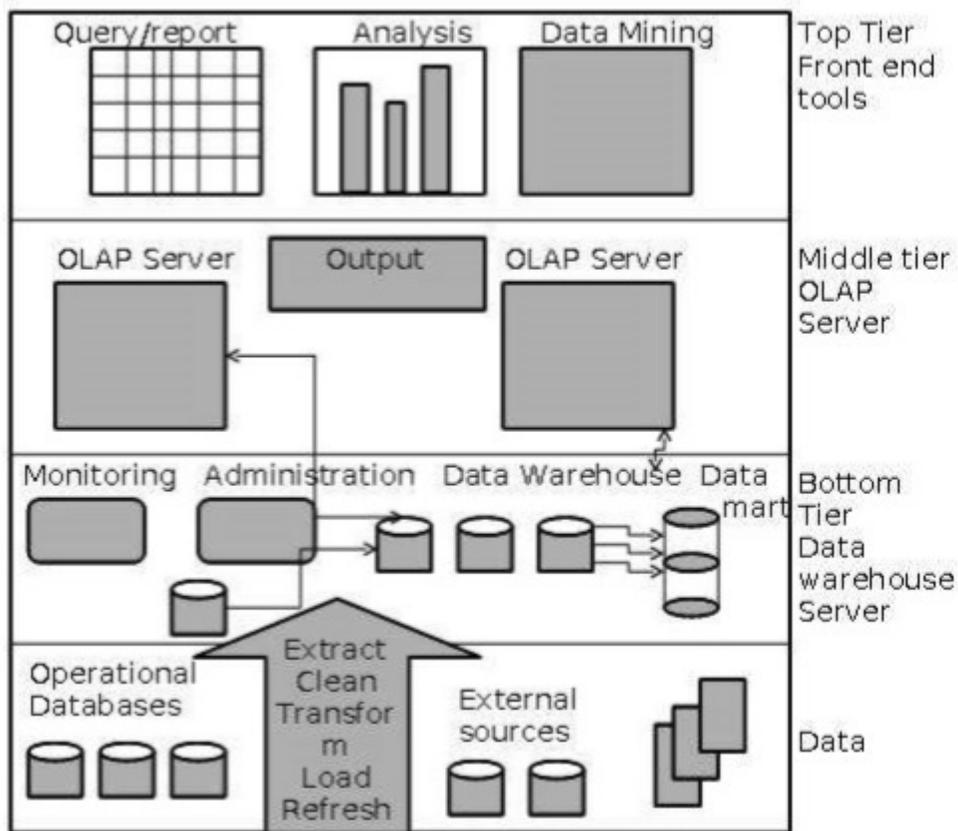
SOLUTION

Q1. Explain the three-tier data warehouse architecture.

Sol. Three-Tier Data Warehouse Architecture

- **Bottom Tier** – The bottom tier of the architecture is the data warehouse database server. It is the relational database system. We use the back end tools and utilities to feed data into the bottom tier. These back end tools and utilities perform the Extract, Clean, Load, and refresh functions.
- **Middle Tier** – In the middle tier, we have the OLAP Server that can be implemented in either of the following ways.
 - By Relational OLAP (ROLAP), which is an extended relational database management system. The ROLAP maps the operations on multidimensional data to standard relational operations.
 - By Multidimensional OLAP (MOLAP) model, which directly implements the multidimensional data and operations.
- **Top-Tier** – This tier is the front-end client layer. This layer holds the query tools and reporting tools, analysis tools and data mining tools.

The following diagram depicts the three-tier architecture of data warehouse –



Q2. Explain various access tools in brief.

Sol: Access tools

The principal purpose of data warehousing is to provide information to business users for strategic decision-making. These users interact with the data warehouse using front-end tools. Although ad hoc requests, regular reports and custom applications are the primary delivery vehicles for the analysis done in most data warehouses, many development efforts of data warehousing projects are focusing on exceptional reporting also known as alerts, which alert a user when a certain event has occurred.

The front-end user tools can be divided into five major groups:

1. Data query and reporting tools.
2. Application development tools.
3. Executive information systems (EIS) tools.
4. Online analytical processing (OLAP) tools.
5. Data mining tools.

Query and reporting tools:

This category can be further divided into two groups: reporting tools and managed query tools. Reporting tools can be divided into production reporting tools and desktop report writers. Production reporting tools let companies generate regular operational reports or support high-volume batch jobs, such as calculating and printing pay cheques. Report writers, on the other hand, are affordable desktop tools designed for end-users. Managed query tools shield end-users from the complexities of SQL and database structures by inserting a metalayer between users and the database.

Application development tools:

The analytical needs of the data warehouse user community exceed the built-in capabilities of query and reporting tools. Organisations will often rely on a true and proven approach of in-house application development, using graphical data access environments designed primarily for client-server environments. Some of these application development platforms integrate well with popular OLAP tools, and can access all major database systems, including Oracle and IBM Informix.

Executive information systems (EIS) tools:

The target users of EIS tools are senior management of a company. The tools are used to transform information and present that information to users in a meaningful and usable manner. They support advanced analytical techniques and free-form data exploration, allowing users to easily transform data into information. EIS tools tend to give their users a high-level summarisation of key performance measures to support decision-making.

OLAP :

These tools are based on concepts of multidimensional database and allow a sophisticated user to analyse the data using elaborate, multidimensional and complex views. Typical business applications for these tools include product performance and profitability, effectiveness of a sales program or a marketing campaign, sales forecasting and capacity planning. These tools assume that the data is organised in a multidimensional model, which is supported by a special multidimensional database or by a Relational database designed to enable multidimensional properties.

Data mining tools:

Data mining can be defined as the process of discovering meaningful new correlation, patterns and trends by digging (mining) large amounts of data stored in a warehouse, using artificial intelligence (AI) and/or statistical/mathematical techniques. The major attraction of data mining is its ability to build predictive rather than retrospective models. Using data mining to build predictive models for decision-making has several benefits. First, the model should be able to explain why a particular decision was made. Second, adjusting a model on the basis of feedback from future decisions will lead to experience accumulation and true organisational learning. Finally, a predictive model can be used to automate a decision step in a larger process.

Q3. Differentiate between OLAP and OLTP.

Sol:

	<i>OLTP System Online Transaction Processing (Operational System)</i>	<i>OLAP System Online Analytical Processing (Data Warehouse)</i>
Source of data	Operational data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP Databases
Purpose of data	To control and run fundamental business tasks	To help with planning, problem solving, and decision support
What the data	Reveals a snapshot of ongoing business processes	Multi-dimensional views of various kinds of business activities
Inserts and Updates	Short and fast inserts and updates initiated by end users	Periodic long-running batch jobs refresh the data
Queries	Relatively standardized and simple queries Returning relatively few records	Often complex queries involving aggregations
Processing Speed	Typically very fast	Depends on the amount of data involved; batch data refreshes and complex queries may take many hours; query speed can be improved by creating indexes
Space Requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP
Database Design	Highly normalized with many tables	Typically de-normalized with fewer tables; use of star and/or snowflake schemas
Backup and Recovery	Backup religiously; operational data is critical to run the business, data loss is likely to entail significant monetary loss and legal liability	Instead of regular backups, some environments may consider simply reloading the OLTP data as a recovery method

Q4. Explain ROLAP server.

Sol: Relational OLAP servers are placed between relational back-end server and client front-end tools. To store and manage the warehouse data, the relational OLAP uses relational or extended-relational DBMS.

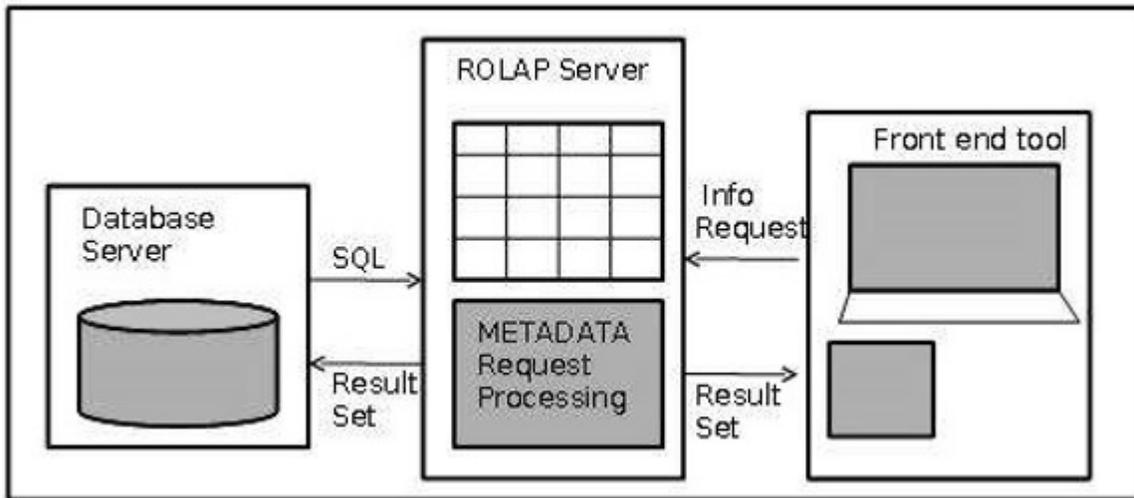
ROLAP includes the following –

- Implementation of aggregation navigation logic
- Optimization for each DBMS back-end
- Additional tools and services

Relational OLAP Architecture

ROLAP includes the following components –

- Database server
- ROLAP server
- Front-end tool.



Advantages:

- Can handle large amounts of data: The data size limitation of ROLAP technology is the limitation on data size of the underlying relational database. In other words, ROLAP itself places no limitation on data amount.
- Can leverage functionalities inherent in the relational database: Often, relational database already comes with a host of functionalities. ROLAP technologies, since they sit on top of the relational database, can therefore leverage these functionalities.

Disadvantages:

- Performance can be slow: Because each ROLAP report is essentially a SQL query (or multiple SQL queries) in the relational database, the query time can be long if the underlying data size is large.
- Limited by SQL functionalities: Because ROLAP technology mainly relies on generating SQL statements to query the relational database, and SQL statements do not fit all needs (for example, it is difficult to perform complex calculations using SQL), ROLAP technologies are therefore traditionally limited by what SQL can do. ROLAP vendors have mitigated this risk by building into the tool out-of-the-box complex functions as well as the ability to allow users to define their own functions.

Government Polytechnic College, Jodhpur
Department of Computer Science(NBA Accredited)

Programme:Diploma

Class Test: III

Term: 2017-18

Course: Cryptography & Network Security

Year: IIIrd

Course CODE: CS-308

Time:09:00 to 10:00

Max.Marks :15

Date:12-04-2018

Instructions to candidates: Attempt all Questions

Note - Mobiles,smart watches or any electronic gadgets are strictly banned.

SI#	Question	Marks	CO (MAPPING)
1	Explain Secure Socket Layer Protocol.	5	CO5
2	Explain Pretty Good Privacy Method.	5	CO5
3	Explain Application Gateway.	5	CO6

Ans. 1

Secure Socket Layer (SSL)

SSL is a widely used general purpose cryptographic system used in the two major Internet browsers: Netscape and Explorer. It was designed to provide an encrypted end-to-end data path between a client and a server regardless of platform or OS. Secure and authenticated services are provided through data encryption, server authentication, message integrity, and client authentication for a TCP connection through HTTP, LDAP, or POP3 application layers. *It was originally developed by Netscape Communications and it first appeared in a Netscape Navigator browser in 1994. It allows for the exchange of multiple messages between two processes and use cryptographic schemes such as digital envelopes, signed certificates, and message digest.*

- a. *SSL is at a lower level in the network stack than S-HTTP, it can work in many other network protocols.*
- b. *SSL is again at a lower level than S-HTTP, it is implemented as a replacement for the sockets API to be used by applications requiring secure*

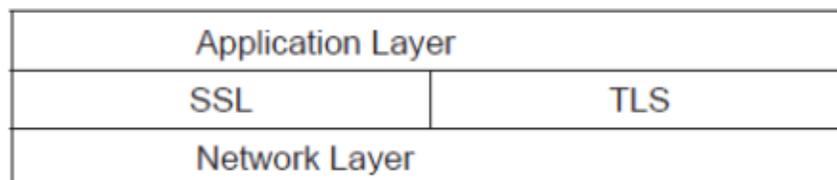


Fig. Transport Layer Security Protocols and Standards

communications. On the other hand, S-HTTP has its data passed in named text fields in the HTTP header. Finally in terms of distribution and acceptance, history has not been so good to S-HTTP. While SSL was released in a free mass circulating browser, the Netscape Navigator, S-HTTP was released in a much smaller and restricted NCSA Mosaic. This unfortunate choice doomed the fortunes of S-HTTP.

SSL Objectives and Architecture

The stated SSL objectives were to secure and authenticate data paths between servers and clients. These objectives were to be achieved through several services that included data encryption, server and client authentication, and message integrity:

- Data encryption – to protect data in transport between the client and the server from interception and could be read only by the intended recipient.

- Server and client authentication – the SSL protocol uses standard public key encryption to authenticate the communicating parties to each other.
- Message integrity – achieved through the use of session keys so that data cannot be either intentionally or unintentionally tampered with.

These services offer reliable end-to-end secure services to Internet TCP connections and are based on an SSL architecture *consisting of two layers: the top layer, just below the application layer, that consists of three protocols, namely the SSL Handshake protocol, the SS Change Cipher Specs Protocol, and the SSL Alert protocol. Below these protocols is the second SSL layer, the SSL Record Protocol layer, just above the TCP layer. See Fig. 17.4.*

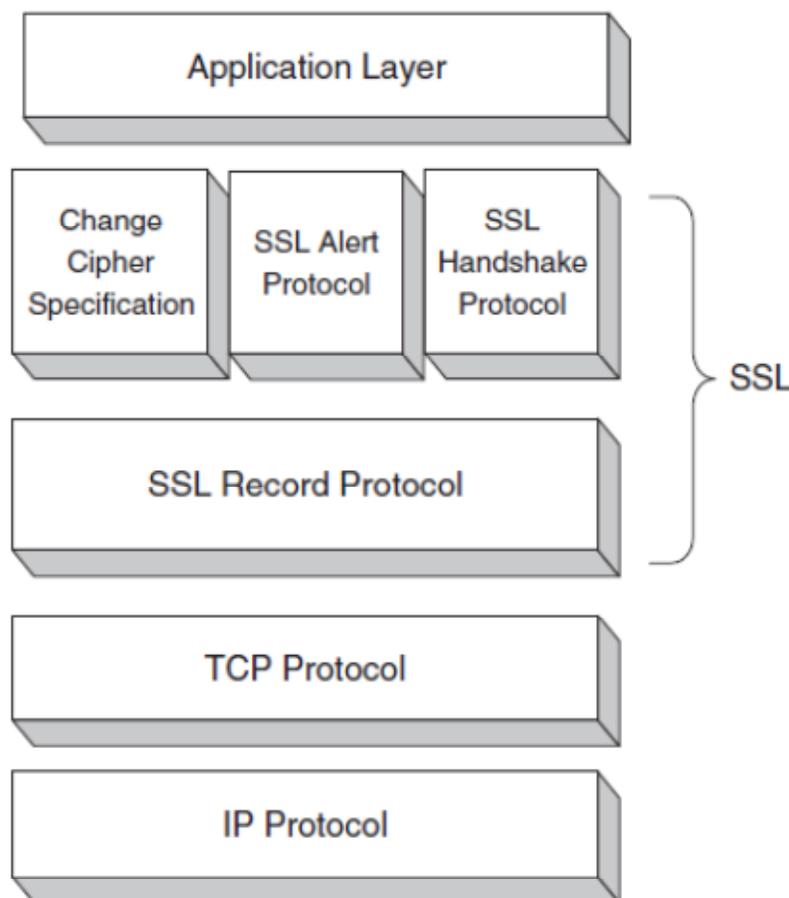


Fig. The SSL Protocol Stack

a. The SSL Handshake

Before any TCP connection between a client and a server, both running under SSL, is established. The SSL Handshake Protocol uses the SSL Record Protocol to exchange a series of messages between an SSL-enabled server and an SSL-enabled client when they first establish an SSL connection.

This exchange of messages is designed to enable the following actions:

- Authenticate the server to the client.
- Allow the client and server to select cryptographic algorithms, or ciphers, they both support.
- Optionally authenticate the client to the server.
- Use public key encryption to generate shared secret keys.
- Establish an encrypted SSL connection.

b. SSL Cipher Specs Protocol

The change cipher spec protocol is used to change the encryption being used by the client and server. It is normally used as part of the handshake process to switch to symmetric key encryption. The CCS protocol is a single message that tells the peer that the sender wants to

change to a new set of keys, which are then created from information exchanged by the handshake protocol.

c SSL Alert Protocol

The SSL Alert Protocol signals problems with an SSL session.

Alert messages convey the severity(strictness) of the message and a description of the alert.

- Upon transmission or receipt of a fatal alert message, both parties immediately close the connection.
- The client and the server must communicate that the connection is ending to avoid a truncation attack.

Either party may initiate the exchange of closing messages.

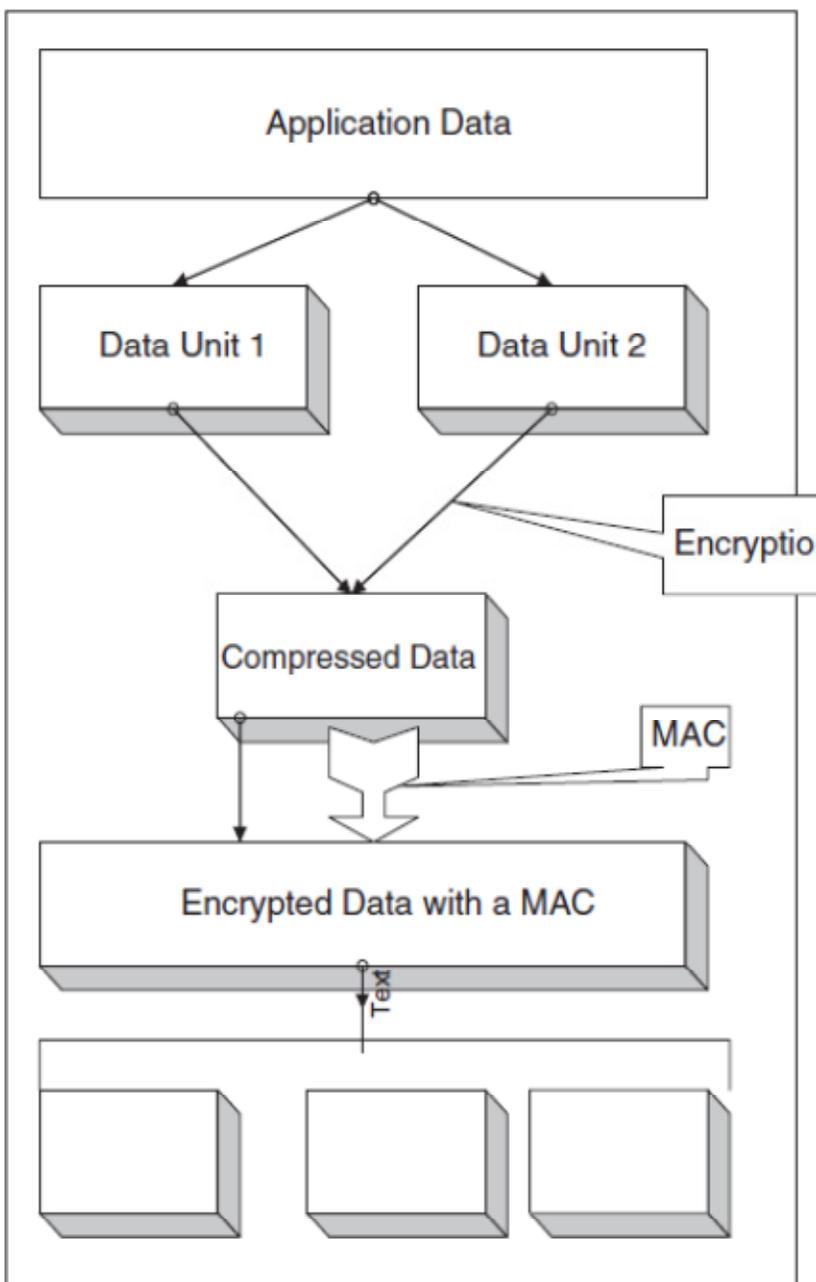
Normal termination occurs when the close_notify message is sent. This message notifies the recipient that the sender will not send any more messages on this connection. The session becomes unresumable if any connection is terminated without a proper close_notify message.

d. SSL Record Protocol

The SSL record protocol provides SSL connections two services: confidentiality and message integrity:

- Confidentiality is attained when the handshake protocol provides a shared secret key used in the conventional encryption of SSL messages.
- Message integrity is attained when the handshake defines a secret shared key used to form a message authentication code (MAC).

In providing these services, the SSL Record Protocol takes an application message to be transmitted and fragments the data that needs to be sent, compresses it, adds a MAC, encrypts it together with the MAC, adds an SSL header, and transmits it under the TCP protocol.



SSL protocols are widely used in all Web applications and any other TCP connections.

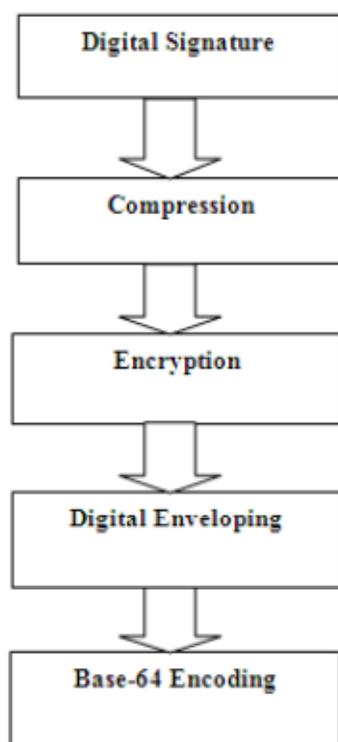
Ans. 2 Pretty Good Privacy (PGP)

Pretty Good Privacy was developed by Phillip Zimmerman to provide a means of secure communication in an insecure electronic environment. PGP is used to encrypt email. PGP's use a public key encryption. The trick is on the receiving end. The only way to open the email is to use the recipients private key. So while a person can send anything to anyone using your public key, only those with the correct private key can read the email.

PGP automatically provides data confidentiality, data integrity, and origin authentication with the option of non-repudiation. PGP provides the following security services:



The following are the main phases of PGP:



Digital Signature: This is a typical process of digital signature. It starts by creating a message digest of the e-mail message using an algorithm such as MD-2 or MD-5. Thus message digest thus created is then encrypted with the senders' private key to form the sender's digital signature.

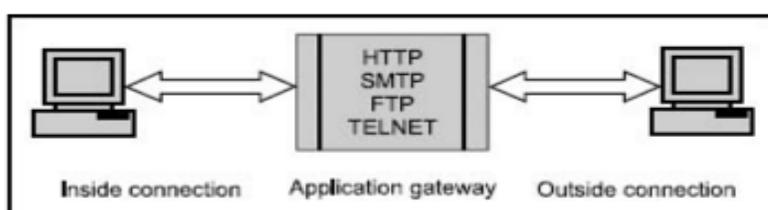
Compression: This is an additional step in PGP. Here the input message as well as the digital signature are compressed together to reduce the size of the final message that will be transmitted. For this the famous zip program is used. ZIP is based on the Lempel-Ziv algorithm.

Digital Enveloping: In this phase, the symmetric key used for encryption in previous phase of encryption is now encrypted with the receiver's public key. The output of the step 3 & step 4 together form a digital envelope.

Base-64 encoding: The output of the step 4 i.e. digital enveloping is encoded.

Ans. 3 Application Gateway

The application gateway is also known as Application Proxy or Proxy server because it acts as proxy or substitute and decides about the flow of application level traffic. This connection provides more security to computers by only displaying the IP address of the firewall or proxy but this decreases the overall speed of sending and receiving information. The following figure shows the application gateway.



Working:

1. An internal user contacts the application gateway using TCP/IP application, such as HTTP or TELNET.

2. The application gateways asks the user about the IP address of the remote host with which the user wants to communicate. It also ask the user id and password required to access the service of application gateway.
3. The user provides the above required information to application gateway.
4. The application gateway now access the remote host on behalf of the user and passes the packets to the remote host.

The communication between a user and the server goes through the proxy server. A proxy server acts as a checkpoint between client and server where requests are validated against specific applications as shown in following figure. A proxy server first checks the users' requests against security policy, if allowable, sends the requests to server. An application proxy inspects not only the IP addresses and Port numbers but also the entire application data portion of an IP packet. Therefore, these firewalls can also prevent the spread of virus.

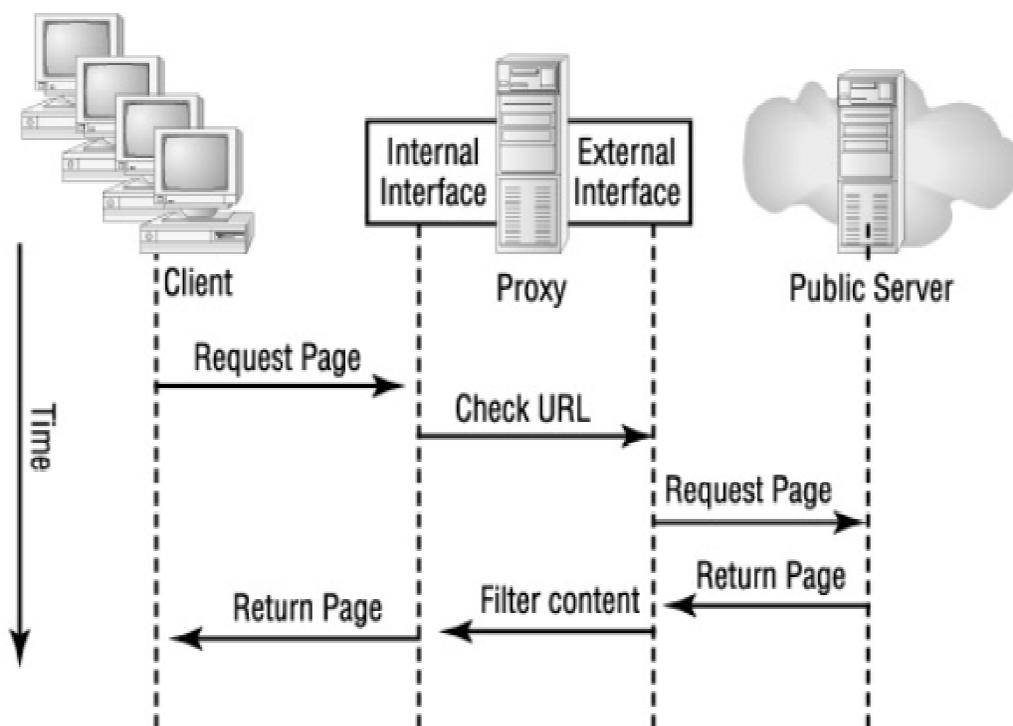


Figure – : Proxy Server Working [60]

A firewall acting as a proxy server performs the following:

- (i) intercepts all packets entering and leaving the protected network.
- (ii) replaces the source IP address with its own IP address to hide the true IP addresses of hosts in protected networks.
- (iii) forwards the packets after checking against security policies and gets the protection of hosts and networks.

Distinction between Packet Filters and Application Proxies

- (i) A packet filter inspects only the packet header but an application proxy inspects the entire application data in the packet.
- (ii) A packet filter allows the same packet to traverse between client and server but an application proxy regenerates an allowed packet as a new packet which is sent from the firewall to the server on the Internet.
- (iii) The application proxy can store the results for example, Web pages, in a cache. Subsequent requests for the same information can be fulfilled from the cache instead of fetching the same content repeatedly. This is not done in packet filter.

Government Polytechnic College, Jodhpur
Department of Computer Science (NBA Accredited)

Programme: Diploma
Course: JAVA TOOLS
Course CODE: CS-309
Max.Marks :15

Class Test: III

Term: 2017-18
Year: IIIrd
Time: 7:45 to 8:45
Date:13-04-2017

Instructions to candidates: Attempt any three Questions

1. Mobiles, smart watches or any electronic gadgets are strictly banned in examination hall.

Sl#	Question	Marks	CO MAPPING
1	Explain JLabel and JComboBox components of swing package.	5	CO5
2	Explain JDBC application architecture in details.	5	CO5
3	Explain Servlet Life Cycle in detail.	5	CO6
4	Briefly explain Generic servlet and Http servlet class.	5	CO6

SOLUTION

Q1. Explain JLabel and JComboBox components of swing package.

Sol: Java JLabel

The object of JLabel class is a component for placing text in a container. It is used to display a single line of read only text. The text can be changed by an application but a user cannot edit it directly. It inherits JComponent class.

JLabel class declaration:

public class JLabel extends JComponent implements SwingConstants, Accessible

Commonly used Constructors:

Constructor	Description
JLabel()	Creates a JLabel instance with no image and with an empty string for the title.
JLabel(String s)	Creates a JLabel instance with the specified text.
JLabel(Icon i)	Creates a JLabel instance with the specified image.
JLabel(String s, Icon i, int horizontalAlignment)	Creates a JLabel instance with the specified text, image, and horizontal alignment.

Commonly used Methods:

Methods	Description
String getText()	It returns the text string that a label displays.
void setText(String text)	It defines the single line of text this component will display.
void setHorizontalAlignment(int alignment)	It sets the alignment of the label's contents along the X axis.
Icon getIcon()	It returns the graphic image that the label displays.
int getHorizontalAlignment()	It returns the alignment of the label's contents along the X axis.

Java JLabel Example

```
import javax.swing.*;
class LabelExample
{
public static void main(String args[])
{
JFrame f= new JFrame("Label Example");
JLabel l1,l2;
l1=new JLabel("First Label.");
l1.setBounds(50,50, 100,30);
l2=new JLabel("Second Label.");
l2.setBounds(50,100, 100,30);
f.add(l1); f.add(l2);
f.setSize(300,300);
f.setLayout(null);
```

```
f.setVisible(true);
} }
```

Java JComboBox

The object of Choice class is used to show popup menu of choices. Choice selected by user is shown on the top of a menu. It inherits JComponent class.

JComboBox class declaration:-

public class JComboBox extends JComponent implements ItemSelectable, ListDataListener, ActionListener, Accessible

Commonly used Constructors:

Constructor	Description
JComboBox()	Creates a JComboBox with a default data model.
JComboBox(Object[] items)	Creates a JComboBox that contains the elements in the specified array.
JComboBox(Vector<?> items)	Creates a JComboBox that contains the elements in the specified Vector.

Commonly used Methods:

Methods	Description
void addItem(Object anObject)	It is used to add an item to the item list.
void removeItem(Object anObject)	It is used to delete an item to the item list.
void removeAllItems()	It is used to remove all the items from the list.
void setEditable(boolean b)	It is used to determine whether the JComboBox is editable.
void addActionListener(ActionListener a)	It is used to add the ActionListener.
void addItemListener(ItemListener i)	It is used to add the ItemListener.

Java JComboBox Example

```
import javax.swing.*;
public class ComboBoxExample {
    JFrame f;
    ComboBoxExample(){
        f=new JFrame("ComboBox Example");
        String country[]={ "India","Aus","U.S.A","England","Newzealand"};
        JComboBox cb=new JComboBox(country);
        cb.setBounds(50, 50,90,20);
        f.add(cb);
        f.setLayout(null);
        f.setSize(400,500);
        f.setVisible(true);
    }
    public static void main(String[] args) {
        new ComboBoxExample();
    }
}
```

Q2. Briefly explain Generic servlet and Http servlet class.

Sol. GenericServlet class

1. GenericServlet class
2. Methods of GenericServlet class
3. Example of GenericServlet class

GenericServlet class implements **Servlet**, **ServletConfig** and **Serializable** interfaces. It provides the implementation of all the methods of these interfaces except the service method.

GenericServlet class can handle any type of request so it is protocol-independent.

You may create a generic servlet by inheriting the GenericServlet class and providing the implementation of the service method.

Methods of GenericServlet class

There are many methods in GenericServlet class. They are as follows:

1. **public void init(ServletConfig config)** is used to initialize the servlet.
2. **public abstract void service(ServletRequest request, ServletResponse response)** provides service for the incoming request. It is invoked at each time when user requests for a servlet.

3. **public void destroy()** is invoked only once throughout the life cycle and indicates that servlet is being destroyed.
4. **public ServletConfig getServletConfig()** returns the object of ServletConfig.
5. **public String getServletInfo()** returns information about servlet such as writer, copyright, version etc.
6. **public void init()** it is a convenient method for the servlet programmers, now there is no need to call super.init(config)
7. **public ServletContext getServletContext()** returns the object of ServletContext.
8. **public String getInitParameter(String name)** returns the parameter value for the given parameter name.
9. **public Enumeration getInitParameterNames()** returns all the parameters defined in the web.xml file.
10. **public String getServletName()** returns the name of the servlet object.
11. **public void log(String msg)** writes the given message in the servlet log file.
12. **public void log(String msg,Throwable t)** writes the explanatory message in the servlet log file and a stack trace.

Example :

```
import java.io.*;
import javax.servlet.*;
```

```
public class First extends GenericServlet{
public void service(ServletRequest req,ServletResponse res)
throws IOException,ServletException{
```

```
res.setContentType("text/html");
```

```
PrintWriter out=res.getWriter();
out.print("<html><body>");
out.print("<b>hello generic servlet</b>");
out.print("</body></html>");
}
}
```

HttpServlet class

1. HttpServlet class
2. Methods of HttpServlet class

The HttpServlet class extends the GenericServlet class and implements Serializable interface. It provides http specific methods such as doGet, doPost, doHead, doTrace etc.

There are many methods in HttpServlet class. They are as follows:

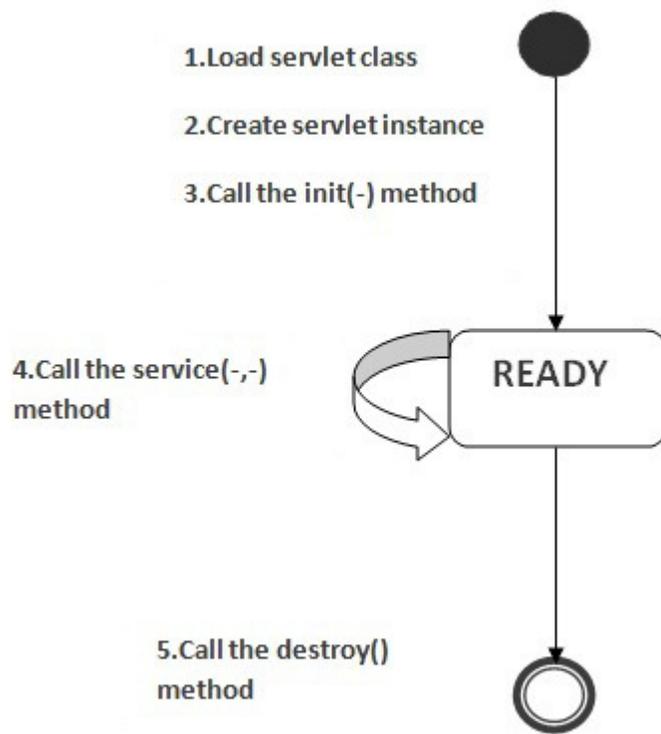
1. **public void service(ServletRequest req,ServletResponse res)** dispatches the request to the protected service method by converting the request and response object into http type.
2. **protected void service(HttpServletRequest req, HttpServletResponse res)** receives the request from the service method, and dispatches the request to the doXXX() method depending on the incoming http request type.
3. **protected void doGet(HttpServletRequest req, HttpServletResponse res)** handles the GET request. It is invoked by the web container.
4. **protected void doPost(HttpServletRequest req, HttpServletResponse res)** handles the POST request. It is invoked by the web container.
5. **protected void doHead(HttpServletRequest req, HttpServletResponse res)** handles the HEAD request. It is invoked by the web container.
6. **protected void doOptions(HttpServletRequest req, HttpServletResponse res)** handles the OPTIONS request. It is invoked by the web container.
7. **protected void doPut(HttpServletRequest req, HttpServletResponse res)** handles the PUT request. It is invoked by the web container.
8. **protected void doTrace(HttpServletRequest req, HttpServletResponse res)** handles the TRACE request. It is invoked by the web container.
9. **protected void doDelete(HttpServletRequest req, HttpServletResponse res)** handles the DELETE request. It is invoked by the web container.
10. **protected long getLastModified(HttpServletRequest req)** returns the time when HttpServletRequest was last modified since midnight January 1, 1970 GMT.

Q3. Explain Servlet Life Cycle in detail.

Sol. Life Cycle of a Servlet (Servlet Life Cycle)

1. Servlet class is loaded
2. Servlet instance is created

3. init method is invoked
4. service method is invoked
5. destroy method is invoked



As displayed in the above diagram, there are three states of a servlet: new, ready and end. The servlet is in new state if servlet instance is created. After invoking the `init()` method, Servlet comes in the ready state. In the ready state, servlet performs all the tasks. When the web container invokes the `destroy()` method, it shifts to the end state.

1) Servlet class is loaded

The classloader is responsible to load the servlet class. The servlet class is loaded when the first request for the servlet is received by the web container.

2) Servlet instance is created

The web container creates the instance of a servlet after loading the servlet class. The servlet instance is created only once in the servlet life cycle.

3) init method is invoked

The web container calls the `init` method only once after creating the servlet instance. The `init` method is used to initialize the servlet. It is the life cycle method of the `javax.servlet.Servlet` interface. Syntax of the `init` method is given below:

```
public void init(ServletConfig config) throws ServletException
```

4) service method is invoked

The web container calls the `service` method each time when request for the servlet is received. If servlet is not initialized, it follows the first three steps as described above then calls the `service` method. If servlet is initialized, it calls the `service` method. Notice that servlet is initialized only once. The syntax of the `service` method of the `Servlet` interface is given below:

```
public void service(ServletRequest request, ServletResponse response)
throws ServletException, IOException
```

5) destroy method is invoked

The web container calls the `destroy` method before removing the servlet instance from the service. It gives the servlet an opportunity to clean up any resource for example memory, thread etc. The syntax of the `destroy` method of the `Servlet` interface is given below:

```
public void destroy()
```

Q4. Explain JDBC application architecture in details.

Sol. JDBC stands for **Java Database Connectivity**, which is a standard Java API for database-independent connectivity between the Java programming language and a wide range of databases.

The JDBC library includes APIs for each of the tasks mentioned below that are commonly associated with database usage.

- Making a connection to a database.

- Creating SQL or MySQL statements.
- Executing SQL or MySQL queries in the database.
- Viewing & Modifying the resulting records.

JDBC is a specification that provides a complete set of interfaces that allows for portable access to an underlying database. Java can be used to write different types of executables, such as –

- Java Applications
- Java Applets
- Java Servlets
- Java ServerPages (JSPs)
- Enterprise JavaBeans (EJBs).

All of these different executables are able to use a JDBC driver to access a database, and take advantage of the stored data. JDBC provides the same capabilities as ODBC, allowing Java programs to contain database-independent code.

JDBC Architecture

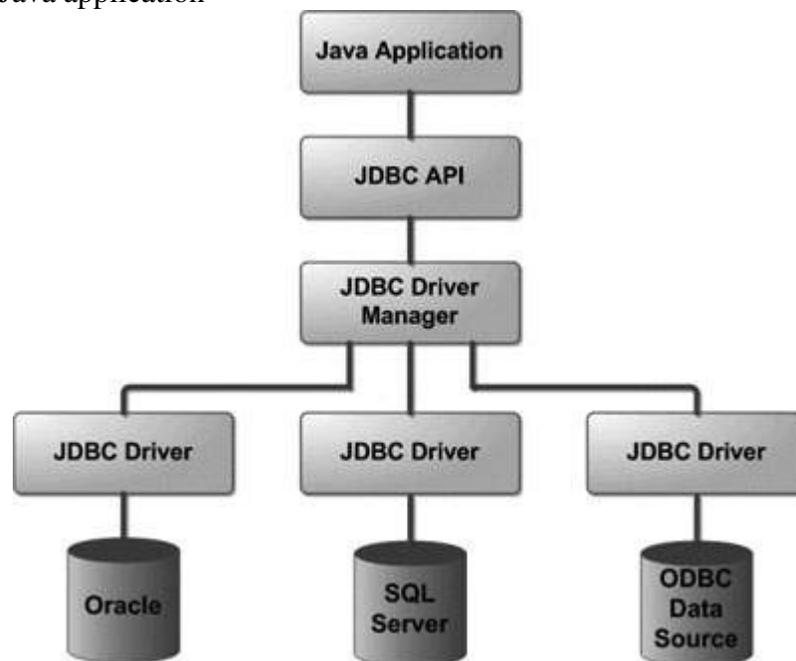
The JDBC API supports both two-tier and three-tier processing models for database access but in general, JDBC Architecture consists of two layers –

- **JDBC API:** This provides the application-to-JDBC Manager connection.
- **JDBC Driver API:** This supports the JDBC Manager-to-Driver Connection.

The JDBC API uses a driver manager and database-specific drivers to provide transparent connectivity to heterogeneous databases.

The JDBC driver manager ensures that the correct driver is used to access each data source. The driver manager is capable of supporting multiple concurrent drivers connected to multiple heterogeneous databases.

Following is the architectural diagram, which shows the location of the driver manager with respect to the JDBC drivers and the Java application –



Common JDBC Components

The JDBC API provides the following interfaces and classes –

- **DriverManager:** This class manages a list of database drivers. Matches connection requests from the java application with the proper database driver using communication sub protocol. The first driver that recognizes a certain subprotocol under JDBC will be used to establish a database Connection.
- **Driver:** This interface handles the communications with the database server. You will interact directly with Driver objects very rarely. Instead, you use DriverManager objects, which manages objects of this type. It also abstracts the details associated with working with Driver objects.
- **Connection:** This interface with all methods for contacting a database. The connection object represents communication context, i.e., all communication with database is through connection object only.
- **Statement:** You use objects created from this interface to submit the SQL statements to the database. Some derived interfaces accept parameters in addition to executing stored procedures.
- **ResultSet:** These objects hold data retrieved from a database after you execute an SQL query using Statement objects. It acts as an iterator to allow you to move through its data.
- **SQLException:** This class handles any errors that occur in a database application.

Programme: Diploma
Course: PHP & MYSQL
Course CODE: CS-310
Max.Marks :15

Class Test: III

Term: 2017-18
Year: IIIrd
Time: 9:00 to 10:00
Date:13-04-2018

Instructions to candidates: Attempt any three Questions

1. Mobiles, smart watches or any electronic gadgets are strictly banned in examination hall.

Sl#	Question	Marks	CO MAPPING
1	What do you mean by Cookies? Explain its purpose and Cookies Myths.	5	CO5
2	Explain the steps for setting and deleting of Cookies in PHP.	5	CO5
3	Explain the process of Database connectivity of PHP with MYSQL.	5	CO6
4	Write PHP script to create one database College having two tables: Employee(emp-id, name, salary, address) and Student(roll-no, name, marks).	5	CO6

Ans. 1

Cookies are small text files stores on the client's computer by the server. Since HTTP is stateless protocol It does not store any information about client. The purposes of the Cookies are as follows:

- (i) Cookies are used to Track or identify the clients.
- (ii) To customize the web pages according to the user's interest.

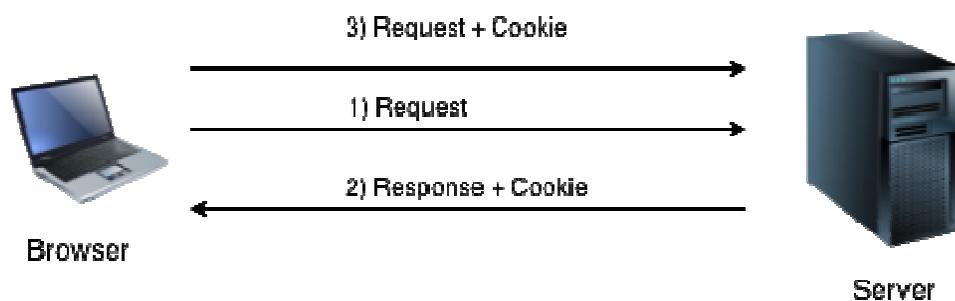


Fig.: Working of Cookies

Cookies Myths

Myth #1: All cookies are malicious. False. Malware and spyware are real internet threats that users need to take seriously; however, as stated above, cookies are not all designed with evil intents. Simplification, speed, and personalization are often their core functionalities. Many viruses are designed to be able to read the cookies on your machines, but they are designed to read everything. Cookies are just one piece of a virus's security breach potential.

Myth #2: Cookies spy on everything you do online. False. Cookies are small stores of data designed for specific functions on specific websites. Not everything is being stored by them. It may appear that you are being followed when ads appear for products you've recently searched for on unrelated sites; however, there's a backend connection at work on such occasions. Cookies can allow targeted ads based on the websites you've visited, but they are only discovering the underlying code of sites pre-designed for such a connection.

Myth #3: Cookies are linked to individuals. False. Cookies can identify a user using a specific browser, but personal identifying information isn't necessarily attached to this user. In fact, if the user were to close one browser (e.g., Chrome) and open another (e.g. Internet Explorer), a cookie could not be 100% sure that these sessions were linked to the same person. Cookies neither follow someone across devices (PC to tablet to phone); nor do they store private information to be recalled at any time and place.

Myth #4: Deleting cookies makes your computer faster. False. Cookies are nearly inconsequential when it comes to speed and space on your computer. Due to browser restrictions, these tiny files are rarely much larger than 4 kilobytes. To give you some perspective, this blog is 15 kilobytes as a MS Word document; a video longer than 10 seconds is usually over 25 megabytes (25000 kb). Sure, cookies take up some space on your machine, but they are hardly the root of memory drain.

Myth #5: Blocking cookies reduces pop-ups. False. Blocking cookies usually has the opposite effect. Many pop-up advertisements use cookies to track whether a user has already seen a certain ad. If a user has already closed a pop-up window, a cookie often stores that information, so that the same ad will not appear repeatedly, annoying the user and thus becoming ineffective. If a user blocks cookies, a website will not know that a viewer has already seen a pop-up and will therefore show the pop-up every single time a user goes to a certain website.

Ans. 2

Setting Cookies with PHP

PHP provided **setcookie()** function to set a cookie. This function requires upto six arguments and should be called before <html> tag. For each cookie this function has to be called separately.

```
setcookie(name, value, expire, path, domain, security);
```

Here is the detail of all the arguments –

- Name – This sets the name of the cookie and is stored in an environment variable called HTTP_COOKIE_VARS. This variable is used while accessing cookies.
- Value – This sets the value of the named variable and is the content that you actually want to store.
- Expiry – This specify a future time in seconds since 00:00:00 GMT on 1st Jan 1970. After this time cookie will become inaccessible. If this parameter is not set then cookie will automatically expire when the Web Browser is closed.
- Path – This specifies the directories for which the cookie is valid. A single forward slash character permits the cookie to be valid for all directories.
- Domain – This can be used to specify the domain name in very large domains and must contain at least two periods to be valid. All cookies are only valid for the host and domain which created them.
- Security – This can be set to 1 to specify that the cookie should only be sent by secure transmission using HTTPS otherwise set to 0 which mean cookie can be sent by regular HTTP.

Following example will create two cookies name and age these cookies will be expired after one hour.

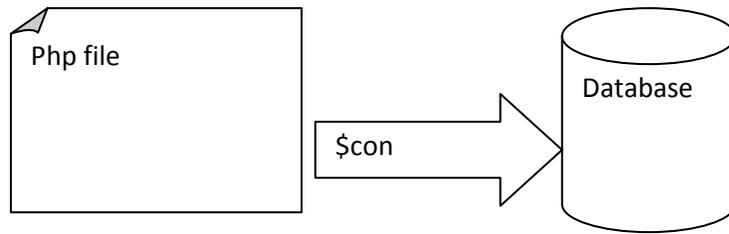
```
<?php
    setcookie("name", "Ram", time()+3600, "/", "", 0);
    setcookie("age", "36", time()+3600, "/", "", 0);
?>
<html>
    <head>
        <title>Setting Cookies with PHP</title>
    </head>
    <body>
        <?php echo "Set Cookies"?>
    </body>
</html>
```

Deleting Cookie with PHP

We can delete a cookie by calling the same setcookie() function with the cookie name and any value (such as an empty string) however this time you need to set the expiration date in the past, as shown in the example below:

```
<?php
    setcookie("name", "", time()- 60, "/", "", 0);
    setcookie("age", "", time()- 60, "/", "", 0);
?>
<html>
    <head>
        <title>Deleting Cookies with PHP</title>
    </head>
    <body>
        <?php echo "Deleted Cookies" ?>
    </body>
</html>
```

Ans. 3 Database Connectivity of php with mysql



Step 1:

- To access the database functionality we have to make a connection to database using php.
- `mysql_connect()` function is used to establish the connection to mysql server.
- Syntax:

```
$con=mysql_connect("localhost","root","");
```

Where, localhost=server name, root= mysql user default, " " = password(it is blank)

Step 2:

The `mysql_select_db()` function is used to display data from a table.

Syntax:

```
mysql_select_db("database name",$con);  
where, $con = connection name
```

Step 3:

The `mysql_query()` function is used for creating query.

Syntax:

```
$con = mysql_query("Query String", $con);
```

Step 4:

The `mysql_fetch_array()` function return the row from the number of records available in the database as an associative array or humaric array.

Syntax:

```
mysql_fetch_array("Query string", $con);
```

Step 5:

After work with the database is done we have to close the connection using `mysql_close()` function, in which the connection to the database is passed.

Syntax:

```
mysql_close($con);
```

Ans. 4

```
<?php  
$con=mysql_connect("localhost","root","");  
if(!$con)  
{  
    Die("Could not Connect!!!");  
}  
$q="create database College";  
mysql_query($q,$con);  
mysql_select_db("College",$con);  
$q="create table Employee(emp-id int, emp-name varchar(20), salary float, address(30)";  
mysql_query($q,$con);  
$q="create table Student(roll-no int, name varchar(20), marks int)";  
mysql_query($q,$con);  
echo "College Database Created with Employee and Student tables";  
mysql_close($con);  
?>
```