

**PART-1**

*Introduction : Introduction and Web Development Strategies, History of Web and Internet, Protocols Governing Web, Writing Web Projects, Connecting to Internet, Introduction to Internet Services and Tools, Introduction to Client-Server Computing.*

**CONCEPT OUTLINE : PART-1**

- WWW stands for World Wide Web. It is an information system where various documents containing information are interlinked together.
- There are various protocols governing web :
 

i. HTTP	ii. OSPF
iii. UDP	iv. FTP
v. ICMP	vi. TCP/IP
vii. POP3	viii. Telnet
ix. RIP	x. IGRP
xi. MIME	xii. Ping
- A website is a related collection of interlinked web pages that includes a beginning file called homepage.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 1.1.** Give the history of World Wide Web.

**Answer**

1. In the history of the World Wide Web, there are two lines to be traced : the development of hypertext, and the development of the internet protocols.
2. In 1972, DARPA starts research leading to the internet.
3. Its main characteristic is the automatic routing of information packets and circumventing the problem of network vulnerability.
4. In 1979, Charles Goldfarb invents SGML.
5. This idea separates content structure from presentation.
6. In 1975, Alan Kay produces the first personal computer.

## Web Technology

1-3Z (CB-S/PT-0)

7. In 1987, CERN and the US laboratories connect to the Internet as the medium of exchanging data between the laboratories.
8. In 1991 SLAC, the Stanford Linear Accelerator Center in California became the first web server in USA.
9. In 1993, the portable browser is released by CERN as Freeweb. Many other browsers are now join with servers.
10. In 1995, Sun Microsystems produces HotJava, a browser which incorporates interactive objects.
11. In 2000, a massive denial of service attack is launched against major websites, including Yahoo, Amazon and eBay.
12. In 2004, Abilene, the Internet backbone, upgraded from 9.6 Gbps to 10 Gbps. Network solutions begin offering 100 year domain registration.

**Que 18:** Discuss web development strategy in brief.

**Answer:**

Web development strategy includes following :

1. **Identify target user:** Identify the user of the website by doing market research.
2. **Make our design portable:** To be successful, website design should be portable and accessible across different browsers, operating systems, environment whether they look same to all their users.
3. **Design for low bandwidth:** Web pages in website should be accessible at any connection speeds. If page download slowly then users will leave the website before they see the content.
4. **Plan for clear presentation and easy access to information:** Presentation of the information on the website must be clear and easily accessible to the user.
5. **Create smooth transitions:** Plan to create a unified look among the sections and pages of site. Before the identifying elements of the site and creates smooth transitions from one page to another.

**Que 19:** What do you understand by web technologies ? Write down the steps to develop multi-department and large scale website.

Q1

[AKTU 2010-14 Marks 07]

What are the technologies used in web design ? What is the role of scripting languages in web design ?

[AKTU 2014-15, Marks 08]

1-4Z (CB-S/PT-0)

Introduction and Core Java

**Answer:**

Web technologies are the technologies which are used to develop website for the World Wide Web or an Internet. Following are various web technologies :

### 1. Client-side technologies :

#### a. XML/HTML :

- i. XML is a markup language designed to structure information for presentation on web pages.
- ii. All XML programs are written to ensure that it is clean, valid and of the highest standard.

#### b. Cascading style sheets : Cascading style sheets control how web pages are displayed in the browser, and allow the separation of presentation from structure and content.

#### c. JavaScript : JavaScript is a lightweight scripting technology which is used alongside XML/HTML documents to make websites more interactive.

### 2. Server-side technologies :

#### a. PHP : PHP is a fast, server-side scripting language that is used to create interactive, dynamic websites.

#### b. CGI/Perl : Perl is a programming language that can handle input and output from a web server, usually through the Common Gateway Interface.

#### c. XML/HTML :

- i. XML is a software and hardware independent markup language designed for describing and transmitting information.
- ii. XSL is a language for defining, transforming and formatting XML documents.

#### d. MySQL :

- i. MySQL is a fast, open-source relational database management system that uses the popular Structured Query Language (SQL).
- ii. MySQL is perfect for most websites that need database functionality.

### 3. Multimedia :

#### a. Macromedia Flash :

- i. Macromedia Flash is a product designed to provide websites with animation, sound, interactivity and database integration.
- ii. Macromedia Flash can create a multimedia website.

**Steps to develop a multi-department and large scale website :**

1. Set-up an "Internet" department.
2. Funding the internet effort.
3. Developing the online strategy of company.
4. Getting to know our audience.
5. Balancing the needs of different units.
6. Meeting the needs of different audiences.

**Role of scripting language in web design :**

1. Scripting language reflects the object orientation of web pages.
2. Scripting language was designed to add interactivity to HTML pages.

**Que 1.4.** | Describe in brief, the growth of the web technology.

Explain the protocols governing the web.

**AKTU 2013-14, Marks 07****Answer**

Growth of web technology : Refer Q. 1.1, Page 1-2Z, Unit-1.

**Protocols governing web :****1. HTTP :**

- a. Hypertext Transfer Protocol (HTTP) is a method used to transfer or convey information on the World Wide Web.
- b. HTTP is a request/response protocol between clients and servers.
- c. The originating client, such as a web browser, spider, or other end-user tools, is referred as the user agent.
- d. The destination server, which stores or creates resources such as HTML files and images, is called server.

**2. ICMP :**

- a. The Internet Control Message Protocol (ICMP) is primarily used by networked computers operating systems to send error messages.
- b. The purpose of these control messages is to provide feedback about problems in the communication environment.

**3. RIP :**

- a. Routing Information Protocol (RIP) is a dynamic routing protocol based on the Bellman-Ford algorithm.
- b. Routing is the method by which the host or gateway decides where to send the datagram.
- c. The goal of RIP is to supply the information that is needed to do routing.

**4. OSPF :**

- a. Open Shortest Path First (OSPF) is classified as an Interior Gateway Protocol (IGP).
- b. This means that it distributes routing information between routers belonging to a single Autonomous System (AS).
- c. OSPF also provides the authentication of routing updates, and utilizes IP multicast when sending/receiving the updates.

**5. TCP/IP :**

- a. TCP/IP stands for Transmission Control Protocol / Internet Protocol.
- b. TCP/IP is the communication protocol for communication between computers on the internet.
- c. TCP is connection oriented protocol.
- d. TCP allows the transmission of arbitrary amount of data by breaking it into stream of separate IP packets.

**6. IGRP :**

- a. The principle goal in creating IGRP (Interior Gateway Routing Protocol) was to provide a robust protocol for routing within an Autonomous System (AS).
- b. IGRP is intended for use in gateways connecting several networks.
- c. IGRP supports multiple metrics for each route, including bandwidth, delay, load, and reliability to compare two routes.

**7. UDP :**

- a. User Datagram Protocol (UDP) is a connectionless protocol without any error detection facility.
- b. The protocol is used for simply transmission of data.
- c. This protocol provides a procedure for application programs to send messages to other programs with a minimum of protocol mechanism.

**Que 1.5.** | What do you mean by protocol ? Provide the name of those protocols which help in web and explain the working of those protocols.**AKTU 2014-15, Marks 03****Answer**

1. A protocol is a set of rules which is used by computers to communicate with each other across a network.
2. A protocol is a standard that controls the connection, communication, and data transfer between computing end points.
3. Protocols may be implemented by hardware, software, or a combination of the two.

Protocols which help in web are : Refer Q. 1.4, Page 1-5Z, Unit-1.

**Que 1.6.** Briefly explain the internet concepts.

**AKTU 2016-17, Marks 10**

OR

What is internet ?

**Answer**

1. Internet is a global system of interconnected computer networks that use the standard Internet Protocols suite (TCP/IP) to serve billions of users worldwide.
2. Internet is a network of networks that consists of millions of private and public, academic, business, and government networks of local to global scope that are linked by a broad array of electronic, wireless and optical networking technologies.
3. The internet carries a vast array of information resources and services, most notably the inter-linked hypertext documents of the WWW and the infrastructure to support electronic mail.
4. Most traditional communications media, such as telephone and television services, are reshaped or redefined using the technologies of the internet, giving rise to services such as Voice over Internet Protocol (VoIP).
5. Newspaper, book and other print publishing has been reshaped into web sites, blogging, and web feeds.
6. The internet has enabled or accelerated the creation of new forms of human interactions through instant messaging, internet forums, and social networking sites.

**Que 1.7.** Why it is important to identify the object in web development strategies ? Also explain, with the help of block diagram, web development process.

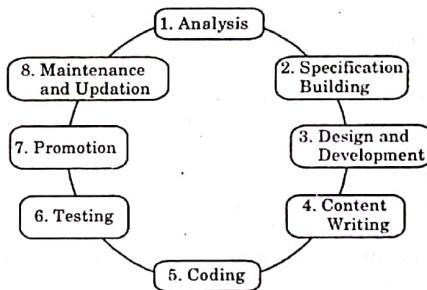
**AKTU 2014-15, Marks 03**

**Answer**

It is essential to identify objects in web development strategies because it will help the developer to understand the elements that play a vital role in business applications.

**Web development process :**

The process of website development can be divided into following different life cycle steps :



**Fig. 1.7.1. Block diagram**

**1. Analysis :**

- a. Once a customer starts discussing his requirements, the team gets towards the preliminary requirement analysis.
- b. The first important thing is finding the targeted audience. Then, all the present hardware, software, people and data should be considered.

**2. Specification building :**

- a. Preliminary specifications are drawn up by covering up each and every element of the requirement.
- b. After reviewing and approving the preliminary document, a written proposal is prepared, outlining the scope of the project including responsibilities, timelines and costs.

**3. Design and development :**

- a. In this step, the layouts and navigation will be designed as a prototype.
- b. Throughout the design phase the team should develop test plans and procedures for quality assurance. It is necessary to obtain client approval on design and project plans.
- c. In parallel, the database team will understand the requirements and develop the database with all the data structures and sample data will also be prepared.

**4. Content writing :** There are professional content developers who write industry specific and relevant content for the site. The grammatical and spelling check should be in this phase.

**5. Coding :**

- a. In coding stage programmers add his code without disturbing the design.
- b. Coding team generate necessary testing plans as well as technical documentation.

- 6. Testing :**
- Both automated testing and manual testing should be done without fail.
  - After doing all the testing, a live testing is necessary for website and web based applications. After uploading the site there should be a complete testing.
- 7. Promotion :**
- Promotion needs preparation of meta tags, constant analysis and submitting the URL to the search engines and directories.
  - The site promotion is normally an ongoing process as the strategy of search engine may change quite often.
- 8. Maintenance and update :** Once the website is operational, ongoing promotion, technical maintenance, content management and staff training is needed on a regular basis.

**Que 1.8.** Explain web project.

OR

What are the stages and strategies required to develop a web project ?

[AKTU 2013-14, Marks 07]

**Answer**

- A web project is the process of developing and creating a website, activities in a network which are aimed at a pre-defined goal.
- The network can be both accessible for everyone, as in the internet, or only for certain people, as in an intranet.

Following are the various stages and strategies required in order to develop a web project :

**Phase-I : Strategy :**

- Goals and objectives
- Team building
- Research and review
- Project proposal

**Phase-II : Design and specification :**

- Developing concepts
- Content planning
- Rough design
- Final design
- Build prototype
- Prototype testing

Phase-III : Production or development

Phase-IV : Testing and maintenance

Phase-V : Register with ISP

Phase-VI : Launch

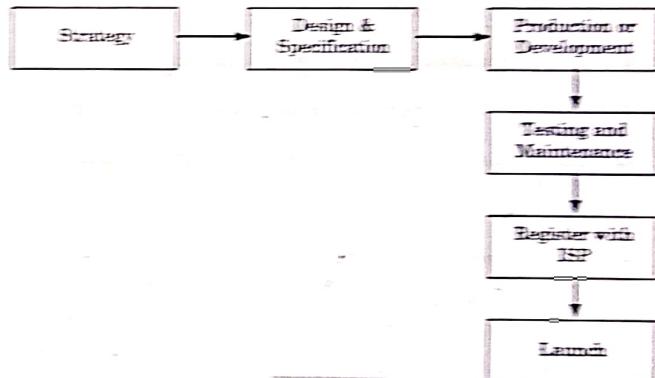


Fig. 1.8.1.

**Que 1.9.** Explain various protocols governing web. Also, explain

[AKTU 2017-18, Marks 03]

**Answer**

Protocols governing web : Refer Q. 1.4, Page 1-5Z, Unit-1.

**Web team :**

- Web team is a group of various technical experts in a developing site from coding the page to maintain the web server.
- The ideal web team consists of three sub-teams :
  - The client-side specialists, who create an attractive, clear front-end.
  - The server-side specialists, who create a smoothly operating back-end.
  - The supporting specialists who make sure that the other two sub-teams can do their jobs.

**Que 1.10.** What is internet service ? Explain various types of internet service.

**Answer**

- Internet service provides a way for data to be transferred from internet servers to our computer.

## Web Technology

### 1-11 Z (CS-5/IT-6)

2. Internet service allows us to access huge amount of information such as text, graphics, sound and software over the internet.
- Four different categories of internet services are as follows :**
1. **Communication services :** There are various communication services available that offer exchange of information with individuals or groups which are as follows :
    - a. **Electronic mail :** Used to send electronic message over the internet.
    - b. **Telnet :** Used to logon to a remote computer that is attached to the internet.
    - c. **Mailing lists :** Used to organize group of internet users to share common information through e-mail.
    - d. **Internet telephony :** It allows the internet users to talk across the internet to any PC equipped to receive the call.
  2. **Information retrieval services :** There exist several information retrieval services offering easy access to information present on the internet which are as follows :
    - a. File Transfer Protocol (FTP)
    - b. Gopher
  3. **Web services :**
    - a. Web services allow exchange of information between applications on the web.
    - b. Using web services, applications can easily interact with each other.
    - c. The web services are offered using concept of Utility Computing.
  4. **World Wide Web (WWW) :**
    - a. WWW is also known as W3.
    - b. WWW offers a way to access documents spread over the several servers over the internet.
    - c. These documents may contain texts, graphics, audio, video and hyperlinks.
    - d. The hyperlinks allow the users to navigate between the documents.

**Que 1.11.** What are the essential skills that must be identified while selecting the members of a web project team ?

AKTU 2014-15, Marks 03

## Answer

The essential skills that must be identified while selecting the members of a web project team are :

1. **Project management skills :** Key project management responsibilities include creating clear and attainable project objectives, building the

### 1-12 Z (CS-5/IT-6)

## Introduction and Core Java

- project requirements, and managing the triple constraint for projects, which is cost, time, and scope.
2. **Information design/architecture skills :** The ability to design a useful user interface that includes how the user will interact and navigate the site.
3. **Graphic design skills :** The graphic designer or graphic artist has the unique job of taking the verbal ideas from the client and developing a creative way to capture the information.
4. **Graphic production skills :** The ability to create web graphics that are fast-loading and that looks great on all browsers.
5. **Content development skills :**
  - a. The ability of researching, writing, gathering, organizing, and editing information for publication on websites.
  - b. Website content may consist of prose, graphics, pictures, recordings, movies or other digital assets that could be distributed by a hypertext transfer protocol server, and viewed by a web browser.
6. **Programming skills :** The ability to create web pages using HTML, JavaScript and other client/server scripting languages.
7. **Technical/network infrastructure skills :** The ability to manage programming and server/network development for the website and to recommend best strategy based on client's needs.

**Que 1.12.** Why planning is must before developing a website ?  
What are the advantages of early planning ?

AKTU 2014-15, Marks 03

## Answer

1. A successful website is a result of successful planning.
2. Hence, before creating and uploading website, it is important to take the time to plan exactly what is needed in the website.
3. Thoroughly considering the audience or target market, as well as defining the purpose and deciding what content will be developed, are extremely important.
4. This will save time, energy and expense in long run. Early planning also helps to maintain our focus. Therefore, planning is must before developing a website.

**Advantages of early planning :** The main advantage of early planning is that planning helps in deciding the following questions :

1. Why are we building the website ?
2. What do we envision as the goal of the site ?
3. What do we hope to gain from creating and maintaining a website ?

4. How will we judge the success of the site ?
5. Who is the target audience ?
6. What are the limiting technical factors affecting our site ?

**Que 1.13.** Discuss the basic elements of a good website design, including navigation considerations. **AKTU 2014-15, Marks 08**

#### Answer

Basic elements of a good website design are as follows :

1. **Original content :** The hallmark of an excellent website is content that clearly states our business in terms that are meaningful to our viewers.
2. **Well-organized and easy to read :** The most important information on any page should be "above the fold" as they say in the business newspaper.
3. **Viewers love to click! Hate to scroll! :** Keep them moving with a site that has many short pages, thoughtfully linked to draw them through our website. This is probably the most powerful, but often overlooked, advantage of using websites to present information.
4. **Share your knowledge :** Every business has knowledge that their viewers might find useful. Use our website to showcase that valuable asset.
5. **Intelligent use of graphics :** Try to balance the graphic elements with the text so that viewer's interest is kept alive and our message gets across.
6. **Be interactive :** Create a site that encourages viewer participation.

Following are the several rules that we need to put in consideration when developing a navigation design :

- a. Keep the design simple
- b. Choose orientation carefully
- c. First finalize the Information Architecture (IA) of website
- d. Keep it consistent
- e. Use specific section names

**Que 1.14.** What is HTTP protocol ? Write its features. Explain HTTP protocol with respect to response and request.

#### Answer

1. The Hypertext Transfer Protocol (HTTP) is an application-level protocol for distributed, hypermedia information systems. This is the foundation for data communication for the World Wide Web (*i.e.*, internet).

2. HTTP is a generic and stateless protocol which can be used for other purposes as well using extensions of its request methods, error codes, and headers.
3. Basically, HTTP is a TCP/IP based communication protocol, that is used to deliver data (HTML files, image files, query results, etc.) on the WWW.
4. The default port is TCP 80, but other ports can be used as well.

#### Features of HTTP protocol :

1. **HTTP is connectionless :** The HTTP client, *i.e.*, a browser initiates an HTTP request and after a request is made, the client disconnects from the server and waits for a response.
2. **HTTP is media independent :** It means, any type of data can be sent by HTTP by specifying appropriate MIME type.
3. **HTTP is stateless :** As HTTP is connectionless and it is a direct result of HTTP being a stateless protocol. The server and client are aware of each other only during a current request.

#### HTTP request :

An HTTP client sends an HTTP request to a server in the form of a request message which includes following format :

- i. A request-line
- ii. Zero or more header (General/Request/Entity) fields followed by CRLF (Carriage Return Line Feed).
- iii. An empty line (*i.e.*, a line with nothing preceding the CRLF) indicating the end of the header fields
- iv. Optionally a message-body

#### HTTP response :

After receiving and interpreting a request message, a server responds with an HTTP response message :

- i. A status-line
- ii. Zero or more header (General/Response/Entity) fields followed by CRLF (Carriage Return Line Feed).
- iii. An empty line (*i.e.*, a line with nothing preceding the CRLF) indicating the end of the header fields
- iv. Optionally a message-body

**Que 1.15.** What is Telnet and remote login ? Explain the working of Telnet tool on the internet.

**Answer****Telnet :**

1. Telnet is a network protocol used on the internet or local area network connections.
2. The purpose of the Telnet protocol is to provide a fairly general, bi-directional, 8-bit byte oriented communications facility.
3. Its primary goal is to allow a standard method of interfacing terminal devices and terminal oriented processes to each other.
4. Telnet clients have been available on most Unix systems for many years and are available virtually for all platforms.
5. Most network equipment and operating systems with a TCP/IP stack support some kind of Telnet service server for their remote configuration.

**Remote login :**

1. When a user wants to access an application program or utility located on a remote machine, the user performs login.

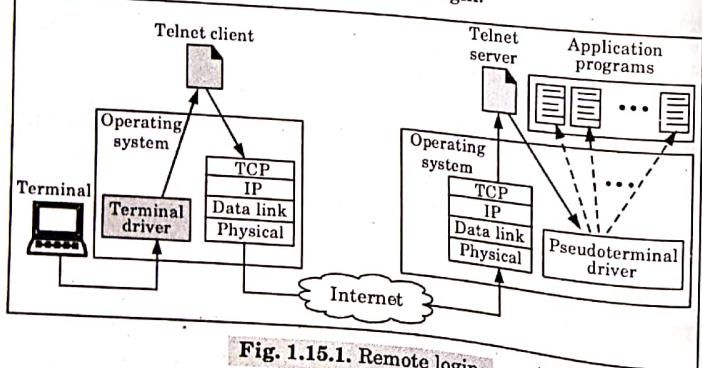


Fig. 1.15.1. Remote login.

2. The Telnet client and server programs come into use.
3. The better solution is a general purpose client/server program that lets a user access any application program on a remote computer.
4. After logging on, a user can use the service available on remote computer and transfer the results back to the local computer.

**Working principle of Telnet :**

1. A Telnet server listens for connections on TCP port 23.
2. When a connection is opened from port 23 and uses a local port above 1023, the remote server will then provide services over that TCP connection.

**1-16 Z (CS-5/IT-6)**

3. When the client receives input from the user, it forwards the information to the Telnet server.
4. When a Telnet connection is first established, each end is assumed to originate and terminate at a "Network Virtual Terminal" or NVT.

**Que 1.16. Explain client-server computing.****Answer**

1. A client-server system is a networked computing model that distributes processes between client and servers.
2. A client-server process usually manages the user-interface portion of the application, validate data entered by the user, dispatch requests to server programs.
3. Client process also manages the local resources that the user interacts with such as the monitor, keyboard, workstation, CPU and other peripherals.

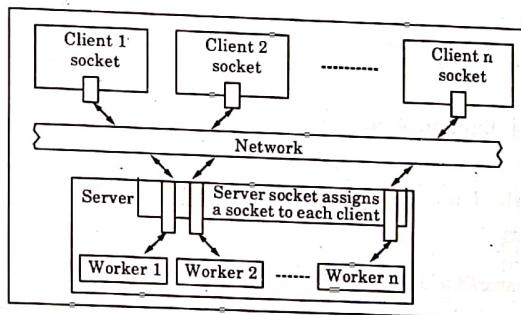


Fig. 1.16.1.

4. Server process fulfills the client request by performing the service requested.
5. The main aim of the server process is to perform the back-end tasks that are common to similar applications.
6. After the server receives requests from clients, it executes database retrieval, updates and manages data integrity and dispatches responses to client requests.

**PART-2**

*Core Java : Introduction, Operator, Data Type, Variable, Arrays, Methods and Classes, Inheritance, Package and Interface.*

**CONCEPT OUTLINE : PART-2**

- Java is a powerful object-oriented programming language.
- Operators in Java are as follows :
  - a. Arithmetic operator
  - b. Increment and decrement operator
  - c. Relational operator
  - d. Boolean operator
  - e. Conditional operator
- Array is a container object that holds a fixed number of values of single data type.
- Java allows classes to inherit commonly used state and behaviour from other classes.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions****Que 1.17.** Explain some features of Java.**OR****Elucidate the features of java.****Answer**

The basic features of Java are given below :

1. **Simple** : All the syntax of Java is like C++. There is no preprocessor and much larger library.
2. **Object-oriented** : All the functions are defined inside the classes. We cannot write any independent function.
3. **Platform independent** : It is platform independent programming language because it compiles to byte-code.
4. **Robust** : Java is graceful in the presence of software or hardware errors. Java has very good exception handlers.
5. **Secure** : Java has no pointers. All pointer-related security problems are gone. It is more secure.
6. **Portable** : Byte-codes are the same regardless of the machine on which it is run, means you can port your program from one system to other system of different architecture.
7. **Multithreaded** : Java programs are capable for easy to set up multiple threads of execution and coordinate parallel processes.

**AKTU 2016-17, Marks 15**

- (CONT'D)
8. **Dynamic** : Because Java programs move through the internet they can exhibit dynamic behaviour.
  9. **Garbage collection** : Java has an automatic garbage collector which releases the objects which are not in use from long time.

**Que 1.18.** Write a short note on operators used in java.**Answer**

An operator is a symbol that usually represents an action or process. These symbols were adapted from mathematics and logic.

Different types of operators are :

1. **Arithmetic operators** : Arithmetic operators include addition, subtraction, multiplication, and division.
2. **Increment and decrement operators** : Adding 1 to a variable is called incrementing that variable, and subtracting 1 is called decrementing. The operators `++` and `--` are called the increment operator and the decrement operator, respectively.
3. **Relational operators** : Relational operators are used to test whether two values are equal, whether one value is greater than another, and so on.
4. **Boolean operators** : In Java, the boolean operators are :
  - a. AND (`&&`)
  - b. OR (`||`)
  - c. NOT (`!`)

These operators combine two boolean values and the result is also a boolean value.

5. **Conditional operator** :

- a. It is a ternary operator i.e., it has three operands and it takes the form :

`<boolean-expression>?<expression-1>:<expression-2>`

**Que 1.19.** Discuss data type in Java.**Answer**

Data type is a classification of data which tells the compiler or interpreter how the programmer intends to use the data.

**Types of data type are :**

- a. **Primitive data types** :

1. Primitive data types are built-in data types.
2. Java initializes all primitive data types to default values.
3. Primitive data types are byte, short, int, long, float, double, boolean and char.

**b. Reference data types :**

1. Reference data types are made by the logical grouping of primitive data types.
2. These are called reference data types because they contain the address of a value rather than the value itself.
3. Reference data types are arrays, objects, interfaces, enum etc.

**Que 1.20.** What is variable ? Describe different types of variable.

**Answer**

1. A variable is a value that can change, depending on conditions or on information passed to the program.
2. There are three ways to declare a variable :
  - a. **Datatype** : The Java Datatype that can be stored in that variable.
  - b. **Variable\_name** : The name of the Java variable.
  - c. **Value** : The initial value stored in the variable.

**Different types of variable are :**

- a. Local variables are defined within a java method, block, or constructor and their scope remains limited to java blocks, classes in Java, and java constructors only.
- b. Instance variables are declared in a class outside any method, constructor, or block, it is of static type. These variables in Java are created when a object is created and destroyed once the object is destroyed.

**For example :**

```
public class Example
{
    //instance variable
    private int var;
    ...
    private void method1()
    {
        //local variable
        int var;
        ...
        this.var ...
    }
    private void method2()
    {
        ...
    }
}
```

**Que 1.21.** What do you mean by arrays in Java ?

**Answer**

1. An array is a container object that holds a fixed number of values of a single type as shown in Fig. 1.21.1.
2. The length of an array is established when the array is created.

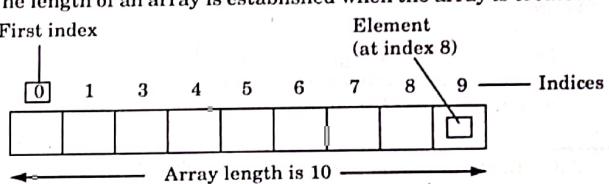


Fig. 1.21.1.

3. Each item in an array is called an element, and each element is accessed by its numerical index.

**Creating, initializing, and accessing an array :**

1. One way to create an array is with the new operator.  
anArray = new int[10]; // create an array of integers
2. The next few lines assign values to each element of the array :  
anArray[0] = 100; // initialize first element  
anArray[1] = 200; // initialize second element
3. Each array element is accessed by its numerical index :  
System.out.println("Element 1 at index 0 :" + anArray[0]);  
System.out.println("Element 2 at index 1 :" + anArray[1]);  
System.out.println("Element 3 at index 2 :" + anArray[2]);

**Que 1.22.** Write a short note on class.

**Answer**

1. A class is a blueprint from which objects are created.
2. A class is a group of objects which have common properties.
3. A class can be declared private or protected.
4. The general syntax of the class construct is :

```
class user_defined_name {
    private:
        data type members ;
        implementation operations ;
        list of friend functions ;
    public:
        data type members ;
        implementation operations ;
```

```

protected:
    data type members ;
    implementation operations ;
};

Example :
class Box {
    public:
        double length; //Length of a box
        double breadth; //Breadth of a box
        double height; //Height of a box
};

```

**Que 1.23.** Explain object and method in Java language.

#### Answer

##### Objects :

1. An object is an instance of a class template.
2. Objects are the basic runtime entities in an object-oriented system.
3. An object has three characteristics :
  - i. **State** : It represents data (value) of an object.
  - ii. **Behaviour** : It represents the behaviour (functionality) of an object.
  - iii. **Identity** : It is a unique ID which is used internally by JVM to identify each object.
4. General syntax to declare object is :

```

Example :
class Test {
    int a, b;
}

class TestObject {
    public static void main (string args []) {
        Test test = new Test ();
    }
}

```

##### Methods :

1. Methods are the set of executable statements.
2. Methods (also called as function) are also the interface to the variables of the class.
3. Methods provide a structured approach to programming. A program can be divided into several methods.

#### Example :

```

public class test {
    public static void welcome () {
        system.out.println ("-----");
    }

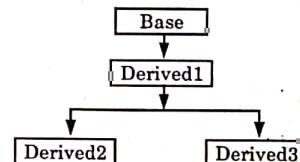
    public static void main(string args []) {
        welcome ();
        system.out.println("Hi !!! This is my first program");
        welcome ();
    }
}

```

**Que 1.24.** Explain the concept of inheritance and its types with suitable example.

#### Answer

1. Inheritance is the mechanism that allows us to extend the definition of a class without making any physical changes to the existing class.
2. Inheritance creates new classes from existing class. Any new class that we create from an existing class is called derived class; existing class is called base class.



**Fig. 1.24.1.**

3. The inheritance relationship enables a derived class to inherit features from its base class. Furthermore, the derived class can add new features of its own.
4. Therefore, rather than creating completely new classes from scratch, we can take advantage of inheritance and reduce software complexity.

#### Types of inheritance :

- a. **Single inheritance** : It is the inheritance hierarchy wherein one derived class inherits from one base class.
- b. **Multiple inheritance** : It is the inheritance hierarchy wherein one derived class inherits from multiple base classes.

- c. **Hierarchical inheritance :** It is the inheritance hierarchy where multiple subclasses inherit from one base class.
- d. **Multilevel inheritance :** It is the inheritance hierarchy where subclass acts as a base class for other classes.
- e. **Hybrid inheritance :** The inheritance hierarchy that reflects any combination of other four types of inheritance.

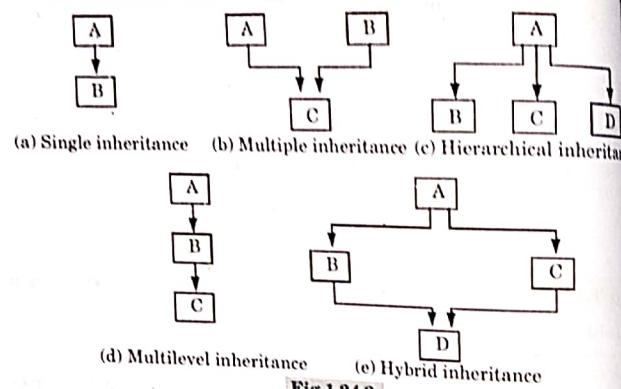


Fig.1.24.2.

**For example :**

```

class First
{
int i = 10;
void inherit()
{
System.out.println("Hello");
}

class Second extends First
{
void inherit()
{
System.out.println(" World");
}

public static void main(String args[])
{
Second s=new Second();
}
  
```

```

s.inherit();
System.out.println(s, b);
}
}
  
```

**Output :**

```

World
10
  
```

**Que 1.25.** Discuss the key features supported by object-oriented programming languages.

**Answer**

Following are the features supported by object-oriented programming languages :

1. **Objects :** Refer Q. 1.23, Page 1-21Z, Unit-1.
2. **Class :** Refer Q. 1.22, Page 1-20Z, Unit-1.
3. **Inheritance :** Refer Q. 1.24, Page 1-22Z, Unit-1.
4. **Polymorphism :**
  - a. Polymorphism is the ability to use an operator or function in different ways.
  - b. An operation may exhibit different behaviour and different instances. The behaviour depends upon the types of data used in the operation.
  - c. Polymorphism plays an important role in allowing objects having different internal structures to share the same external interface.
  - d. This means that a general class of operations may be accessed in the same manner even though specific actions associated with each operation may differ.
  - e. Polymorphism is extensively used in implementing inheritance.
5. **Encapsulation :**
  - a. Encapsulation is the wrapping up of data and methods into single unit called class.
  - b. Using the method of encapsulation, the programmer cannot directly access the data. Data is only accessible through the function present inside for the class.
  - c. Data encapsulation is an important concept of data hiding.
  - d. Data hiding is the implementation details of a class that are hidden from the user.
  - e. The concept of restricted access helps the programmers to write specialized functions or methods for performing the operations on hidden member of the class.

- f. The concept of encapsulation shows that a non-member function cannot access an object's private or protected data.

**Que 1.26.** What are packages in java? How a user-defined package is created in java, explain with example?

AKTU 2017-18, Marks 10

**Answer**

1. Package is a mechanism to encapsulate a group of classes, interfaces and subpackages.
2. Packages are the way to organize files into different directories according to their functionality, usability as well as category.
3. Packages provide a way to hide classes thus preventing other programs or packages from accessing classes that are meant for inter use only.
4. Packages also provide a way for separating "design" from "coding".
5. There are two types of packages in Java :
  - i. **User-defined package** : The package we create is called user defined package.
  - ii. **Built-in package** : The already defined package like `java.io.*`, `java.lang.*` etc are known as built-in packages.

**To create user-defined package :**

User-defined package is created with the help of "package" keyword, and to use a package we use the import keyword.

**Example :**

```

Demo.java
package abhi;
public class Demo
{
    public void sum(int num1,int num2)
    {
        int result;
        result=num1+num2;
        System.out.println("the sum of two numbers is:"+result);
    }
}

Tester.java
import abhi.Demo;
class Tester extends Demo
{
}
  
```

```

public static void main(String args[])
{
    Tester obj=new Tester();
    obj.sum(10,20);
}
  
```

**Que 1.27.** Write a short note on interface with example.

**Answer**

1. An interface defines a set of methods but does not implement them.
2. A class that implements the interface agrees to implement all of the methods defined in the interface.
3. An interface is a collection of method declarations (without definitions). An interface can also include constant declarations.

**Defining an interface :**

1. An interface definition has two components :
  - a. **Interface declaration** : The interface declaration declares various attributes about the interface such as its name and whether it extends another interface.
  - b. **Interface body** : The interface body contains the constant and method declarations within interface body as shown in Fig 1.27.1.

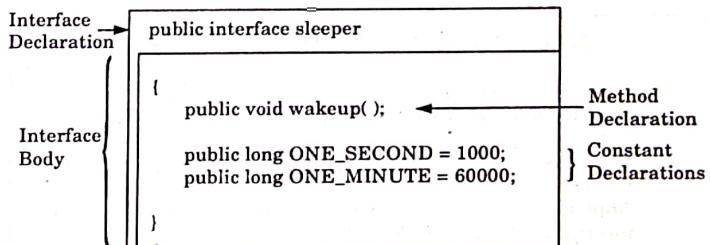


Fig. 1.27.1.

**For example :**

```

// simple interface
interface Calc
{
    void sum(int a, int b);
    void sub(int a, int b);
}

// class implements with Interface
class MyCalc implements Calc
  
```

```

public void sum(int a, int b) // method of Calc
{
    System.out.println("Sum is =" + (a + b));
}
public void sub(int a, int b) // method of Calc
{
    System.out.println("Sub is =" + (a - b));
}
public static void main(String args[])
{
    MyCalc cal = new MyCalc();
    cal.sum(100, 20);
    cal.sub(100, 20);
}

```

**Output :**  
Sum is = 120  
Sub is = 80

**PART-3**

*Exception Handling, Multi-thread Programming, I/O, Java Applet, String Handling, Event Handling, Introduction to AWT, AWT Control, LayoutManagers.*

**CONCEPT OUTLINE : PART-3**

- Exception is an unusual condition that can occur in the program.
- Thread is a small program running continuously. It is a lightweight process.
- Stream is basically a channel on which the data flow from sender to receiver.
- Applets are the small Java programs that can be used in internetworking environment.
- Event means any activity that interrupts the current ongoing activity.
- The graphics programming in Java is supported by the AWT package.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 1.28.** What are exceptions and how they are handled in java ? Explain the keywords try, catch, throw, and finally with example.

**AKTU 2017-18, Marks 10**

**OR**

Explain exception handling in Java.

**Answer**

An exception is an unwanted or unexpected event, which occurs during the execution of a program i.e., at runtime, that disrupts the normal flow of the program's instruction.

**Exception handling :**

1. Exception handling provides a type-safe, integrated approach for handling unusual problems that arise while executing a program.
2. To handle the exceptions, exception handling mechanism is designed and it handles only synchronous errors.
3. The mechanism suggests a separate error handling code that performs the following tasks :
  - a. Find the problem (Hit the exception)
  - b. Inform that an error has occurred (Throw the exception)
  - c. Receive the error information (Catch the exception)
  - d. Take corrective actions (Handle the exception)

**Java exception handling is managed by the following keywords :**

**1. Try :**

- a. Java uses keyword "try" to preface a block of code that is likely to cause an error condition and "throw" an exception.
- b. The try block can have one or more statements that could generate an exception.

```

try
{
    statement; // generates an exception
}
catch (Exception_type e)
{
    statement; // processes the exception
}

```

- c. If any one statement generates an exception, the remaining statements in the block are skipped and execution jumps to the catch block that is placed next to the try block.

- d. Every try statement should be followed by at least one catch statement; otherwise compilation error will occur.
- 2. Catch :**
- A catch block defined by the keyword "catch" catches the exception thrown by the try block and handles it appropriately. The catch block is added immediately after the try block.
  - The catch block can have one or more statements that are necessary to process the exception.
  - The catch statement works like a method definition.
  - The catch statement is passed as a single parameter, which is reference to the type of exception object thrown by the try block.
  - If the catch parameter matches with the type of exception object, then the exception is caught and statements in the catch block will be executed.
- 3. Finally :**
- Java supports another statement known as finally statement that can be used to handle an exception that is not caught by any of the previous catch statements.
  - Finally block can be used to handle any exception generated within a try block.
  - It may be added immediately after the try block or after the last catch block as follows :
- ```

try {
    .....
}
finally {
    .....
}
try {
    .....
}
catch (...) {
    .....
}
catch (...) {
    .....
}
finally {
    .....
}
    
```

- 4. Throw :**
- Java supports "throw keyword" which is used if we want to throw our own exceptions.
  - We can do this by using the keyword throw as follows :  
throw new Throwable\_subclass;

**For example :**

```

throw new ArithmeticException();
throw new NumberFormatException();
class TestFinallyBlock
{
    public static void main(String args[])
    {
        try
        {
            int data=25/0;
            System.out.println(data);
        }
        catch(ArithmeticException e)
        {
            System.out.println("Caught: "+e);
        }
        finally
        {
            System.out.println("finally block is always executed");
        }
        System.out.println("rest of the code...");
    }
}
    
```

**Output :**

Caught: java.lang.ArithmetricException: / by zero  
 finally block is always executed  
 rest of the code...

**Que 1.29. Explain multithreading in Java.****Answer**

- A multithreaded program contains two or more parts that can run concurrently.
- Each part of such a program is called a thread, and each thread defines a separate path of execution.
- Each thread runs parallel to each other.
- Java provides built-in support for multithreaded programming.
- A multithreading is a specialized form of multitasking.

6. In multithread program, each thread has its own life cycle. The life cycle of thread is shown in Fig. 1.29.1.

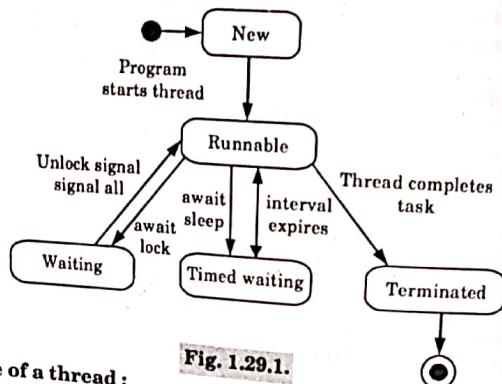


Fig. 1.29.1.

**Life cycle of a thread :**

A thread goes through various stages in its life cycle which are as follows

- a. **New :**
  - i. A new thread begins its life cycle in the new state. It remains in this state until the program starts the thread.
  - ii. It is also referred to as a new born thread.
- b. **Runnable :**
  - i. After a newly born thread is started, the thread becomes runnable.
  - ii. A thread in this state is considered to be executing its task.
- c. **Waiting :**
  - i. Sometimes a thread gets transitions to the waiting state while the thread waits for another thread to perform a task.
  - ii. A thread transitions get back to the runnable state only when another thread signals the waiting thread to continue executing.
- d. **Timed waiting :**
  - i. A runnable thread can enter the timed waiting state for a specified interval of time.
  - ii. A thread in this state transitions back to the runnable state when that time interval expires or when the event it is waiting for, occurs.
- e. **Terminated :** A runnable thread enters the terminated state when it completes its task, otherwise it terminates.

**Que 1.30.** Write a short note on Java I/O.

#### Answer

1. Java programs perform I/O through streams.
2. A stream is linked to a physical device by the Java I/O system to make input and output operation in Java.

- 3. Streams are clean way to deal with I/O without understanding every part of our code and the difference between a keyboard and a network.
- 4. Java defines two types of streams :
  - a. **Byte streams :**
    - i. It provides a convenient means for handling input and output of bytes.
    - ii. Byte streams are used for reading or writing binary data.
    - iii. FileInputStream and FileOutputStream are the most frequently used classes in byte stream.
  - b. **Character streams :**
    - i. It provides a convenient means for handling input and output of characters.
    - ii. They use unicode and therefore, can be internationalized.
    - iii. FileReader and FileWriter are most frequently used classes of character stream.
- 5. An input stream can abstract many different kinds of input from a disk file, a keyboard, or a network socket.
- 6. An output stream may refer to the console, a disk file, or a network connection.

**Que 1.31.** What is applet ? Explain life cycle of applet.

OR

Explain Java applet with example.

#### Answer

1. Applet is a Java program that can be embedded into a web page.
2. It runs inside the web browser and works at client side.
3. Applet is embedded in a HTML page using the APPLET or OBJECT tag . and hosted on a web server.
4. Applets are used to make the website more dynamic and entertaining.
5. All applets are sub-classes (either directly or indirectly) of java.applet.Applet class.
6. Applets can run within a web browser or an applet viewer (standard applet viewer tool).

**Life cycle of an Applet :**

Life cycle of an applet use five methods which are as follows :

- a. **init() :** This method is intended for whatever initialization is needed for our applet.
- b. **start() :** This method is automatically called after the browser calls the init method.
- c. **stop() :** This method is automatically called when the user moves off

- the page on which the applet sits.
- destroy()**: This method is only called when the browser shuts down normally.
  - paint()**: Invoked immediately after the start() method, and also any time the applet needs to repaint itself in the browser.

**Example :**

```
import java.applet.*;
import java.awt.*;
public class our_Applet extends Applet
{
    string disp_str;
    public void init()
    {
        disp_str = "Hello World!";
    }
    public void paint(Graphics g)
    {
        g.drawString(disp_str, 50, 25);
    }
}
```

**Que 1.32.** Explain string handling in Java with example.**Answer**

- String handling is a process of performing different operation such as concatenation, comparison on the string.
- Following are the method used in string handling :
  - Java string length()** : The Java string length() method returns the length of the string. It returns count of total number of characters present in the string.

**For example :**

```
public class Example{
    public static void main(String args[])
    {
        String s1="hello";
        String s2="whatsup";
        System.out.println("string length is: "+s1.length());
        System.out.println("string length is: "+s2.length());
    }
}
```

**Output :**

String length is : 5

- String length is : 7
- Java string compareTo()** : The Java string compareTo() method compares the given string with current string. It returns positive number, negative number or zero.

**For example :**

```
public class CompareToExample{
    public static void main(String args[])
    {
        String s1="hello";
        String s2="hello";
        String s3="hemlo";
        String s4="flag";
        System.out.println(s1.compareTo(s2));
        System.out.println(s1.compareTo(s3));
        System.out.println(s1.compareTo(s4));
    }
}
```

**Output :**0  
-1  
2

- Java string concat()** : The Java string concat() method combines a specific string at the end of another string and ultimately returns a combined string.

**For example :**

```
public class ConcatExample{
    public static void main(String args[])
    {
        String s1="hello";
        s1=s1.concat("how are you ?");
        System.out.println(s1);
    }
}
```

**Output :**

hello how are you ?

- Java string replace()** : The Java string replace() method returns a string, replacing all the old characters or CharSequence to new characters.

**For example :**

```
public class ReplaceExample1{
    public static void main(String args[])
    {
        String s1="hello how are you ?";
        String replaceString=s1.replace('h','t');
        System.out.println(replaceString);
    }
}
```

**Output :**

tello tow are you ?

- Java string equals()** : The Java string equals() method compares the two given strings on the basis of content of the string. If all the characters are matched, it returns true else it will return false.



**For example :**

```
public class EqualsExample{
    public static void main(String args[]){
        String s1="hello";
        String s2="hello";
        String s3="hi";
        System.out.println(s1.equalsIgnoreCase(s2));
        System.out.println(s1.equalsIgnoreCase(s3));
    }
}
```

**Output :**

true  
true  
false

- f.** **Java string contains()** : The Java string contains() method searches the sequence of characters in the string. If the sequences of characters are found, then it returns true otherwise returns false.

**For example :**

```
class ContainsExample{
    public static void main(String args[]){
        String name=" hello how are you doing ?";
        System.out.println(name.contains("how are you"));
        System.out.println(name.contains("hello"));
        System.out.println(name.contains("fine"));
    }
}
```

**Output :**

true  
true  
false

**Que 1.33.** Explain event handling in brief.**Answer**

- Event handling is the mechanism that controls the event and decides what should happen if an event occurs.
- This mechanism has the code which is known as event handler that is executed when an event occurs.
- The modern approach of handling events is based on the delegation event model, which defines standard and consistent mechanisms to generate and process events. Its concept are :
  - A source generates an event and sends it to one or more listeners.
  - In this scheme, the listener simply waits until it receives an event.
  - Once event received, the listener processes the event and then returns.



- The advantage of this design is that the application logic that processes events is clearly separated from the user interface logic generated by those events.
  - A user interface element is able to "delegate" the processing of an event to a separate piece of code.
  - In the delegation event model, listeners must register with a source in order to receive an event notification.
  - This provides an important benefit : notifications are sent only to listeners that want to receive them.
  - Java also allows you to process events without using the delegation event model. This can be done by extending an AWT component.
4. In the delegation event model, an event is an object that describes a state change in a source. It can be generated as a consequence of a person interacting with the elements in a graphical user interface :
- Some of the activities that cause events to be generated are pressing a button, entering a character via the keyboard, selecting an item in a list, and clicking the mouse.
  - Events may also occur that are not directly caused by interactions with a user interface.
  - We are free to define events that are appropriate for our application.
5. A source is an object that generates an event. This occurs when the internal state of that object changes in some way. Sources may generate more than one type of event.
- A source must register listeners in order for the listeners to receive notifications about a specific type of event.
  - Each type of event has its own registration method.
  - The general form of registration is : `public void addTypeListener(TypeListener el)`

**Que 1.34.** Write short note on AWT.**Answer**

- The AWT stands for Abstract Window Toolkit.
- AWT is a library of class which provides GUI tools to develop GUI application and applet.
- It provides many classes for programmers to use. It is the connection between our application and the native GUI.
- It is a Java package and can be used in any Java program by importing `java.awt.*` via the import keyword.

5. It contains three kinds of classes :
- Containers class : Frame, Dialog, Panel, Applet etc.
  - Components class : TextField, Button, Checkbox, Scrollbar, Label, List etc.
  - Custom graphics class : Colour, Font, Dimensions etc.
6. The AWT supports following types of controls :
- Buttons
  - Checkbox
  - CheckboxGroup
  - Choice
  - Label
  - List
  - Scrollbar
  - TextField
  - TextArea

**Que 1.35.** Mention various AWT controls. Explain any four of them.

#### Answer

**AWT controls :** AWT controls are components that allow a user to interact with our application.

**Various AWT controls are as follows :** Refer Q. 1.34, Page 1-36Z, Unit-1.

- Canvas :** A canvas is a graphical component representing a region where we can draw things such as rectangles, circles, and text strings.
- Checkbox :**
  - A Checkbox is a label with a small push button.
  - The state of Checkbox is either true (button is checked) or false (button not checked).
  - The default initial state is false.
- CheckboxGroup :**
  - A CheckboxGroup is used to control the behaviour of a group of Checkbox objects (each of which has a true or false state).
  - Exactly one of the Checkbox objects is allowed to be true at one time.
  - Checkbox objects controlled with a CheckboxGroup are usually referred to as "radio buttons".
- TextArea :**
  - A TextArea is a multi-row text field that displays a single string of characters, where newline ('\n' or '\r' or '\n\r' or '\r\n', depending on platform) ends each row.

- b. The width and height of the field is set at construction, but the text can be scrolled up/down and left/right.

**Que 1.36.** Explain LayoutManager in brief.

**OR**

**What is LayoutManager ? What are the various types of LayoutManager ?**

#### Answer

**LayoutManager :**

- LayoutManager is abstract class, we cannot use it directly.
- LayoutManager class describes how components are "laid out" within a container.
- We must subclass it and provide our own functionality or use a derived class of LayoutManager already created for us.
- To use a layout we must call setLayout( ) for the container with an instance of a LayoutManager.

**Types of LayoutManager :**

- BorderLayout :** This scheme lays out the component in five ways :
  - North-Northern part of the container
  - South-Southern part of the container
  - East-Eastern part of the container
  - West-Western part of the container
  - Center-centered in the container
- CardLayout :** Allows for what Windows programmers have called "tabbed dialogs" or dynamic dialogs.
- GridLayout :** Allows for the layout of components in a grid-like fashion rather than "North" or "Center".
- FlowLayout :** Allows for component to be laid out in a row(or flow) and aligned(left, right, center).
- None :** No layout, the container will not attempt to reposition the components during a update.

#### VERY IMPORTANT QUESTIONS

**Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.**

**PART-1**

**Web Page Designing : HTML : List, Tables, Images, Frames, Forms, CSS.**

**CONCEPT OUTLINE : PART-1**

- HTML stands for Hypertext Markup Language. It is a language interpreted by a browser.
- <font> tags are used for changing font style in HTML document.
- To make a hyperlink text we use <a> tag.
- Lists are used to present the text in the form of list in HTML document.
- Form is a typical layout on the web page by which user can interact with web page.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 2.1.** What is HTML ? Explain the structure of HTML.

**Answer**

1. HTML is a set of special codes that can be embedded in text to add formatting and linking information.
2. HTML is the language interpreted by a browser. Web pages are also called HTML documents.

**Structure of a HTML program :**

1. Every HTML program has a rigid structure.
2. The entire web page is enclosed within <HTML>---</HTML> tags.
3. Within these tags two distinct sections are created using the <HEAD>---</HEAD> tags and the <BODY>---</BODY> tags.
4. An HTML document is divided in two parts :
  - a. **Document head :**
    - i. This HTML tag is used for the identification of the heading or title of HTML document.
    - ii. All the information placed within the <HEAD>---</HEAD> tags is not displayed in browser.

- iii. The HTML tags used to indicate the start and end of the head section are :

<HEAD> <TITLE> --- </TITLE> </HEAD>

**b. Document body :**

- i. This tag is used for indicating the actual content of the HTML documents layout and structure.
- ii. The tags used to indicate the start and end of the main body of text information are :

<BODY>  
:  
</BODY>

- iii. The attributes that the <BODY> tag takes are : BGCOLOR, BACKGROUND, TEXT etc.

**Que 2.2.** Define HTML tags. Also, explain different tags used for text formatting in HTML with example.

**Answer****HTML tags :**

1. Tags are instructions that are embedded directly into the text of the document.
2. HTML is specified as tags in an HTML document i.e., the web page.

**HTML tags are of two types :****1. Paired tags :**

- a. Paired tag is a combination of two tags, opening tag (<B>) and closing tag (</B>).
- b. The opening tag activates the effect and the closing tag turns the effect off.

For example : <HTML>---</HTML> tags.

**2. Singular tags :**

- a. Singular tag is a standalone tag.
- b. It does not have a closing tag.

For example : <BR> tag is used for single line break. Generally, it is kept at the end of every line.

**Some of the tags used for text formatting in HTML are :****1. Title :**

- a. An HTML document has a title that describes what the page is about. This can be achieved by using TITLE tag.
- b. Text included between the <TITLE>---</TITLE> tags shows up in the title bar of the browser window.

**Example :**

<TITLE> Quantum Page </TITLE>

**2. Footer :**

- a. Certain information is needed to be placed at the foot of the document like copyright information, contact details etc. This can be achieved by using <ADDRESS> ----- </ADDRESS> tag.
- b. This tag should ideally be placed immediately after the textual material of the web page.

**Example :**

<ADDRESS> Copyright © QUANTUM PUBLICATIONS  
ADDRESS>

**3. Paragraph breaks :**

- a. <P> tag is used to break paragraph or skip one line between previous line and new line.
- b. <P> tag can be neglected at the end of paragraph.
- c. ALIGN attribute is used in <P> tag with three different values CENTER, LEFT, RIGHT.

**Example :** <P ALIGN = "LEFT">

**4. Line breaks :**

- a. <BR> tag is a standalone tag.
- b. <BR> tag is used to move the text to start from a new line and continue on the same line.
- c. Browsers recognize multiple consecutive <BR> tags.

**Example :** <P> For further details contact us at <BR> QUANTUM PUBLICATIONS<BR> GHAZIABAD<BR> UTTAR PRADESH

**Output :** For further details contact us at

QUANTUM PUBLICATIONS  
GHAZIABAD  
UTTAR PRADESH

**5. <DIV> tag :**

- a. <DIV> tag specifies a particular section in a HTML document.
- b. The possible attribute values for DIV tag are same as paragraph attributes.

**Example :** <DIV ALIGN = "LEFT">  
text  
</DIV>

Que 2.3.

Describe tags used for styling in HTML with examples.

**Answer**

Tags used for styling in HTML are :

**1. Heading styles :**

- a. HTML supports six different levels of headings.
- b. The highest level header format is <H1> and the lowest level is <H6>.
- c. As the number next to <H> (1, 2,.....) increases, the font size decreases.

**Example :**

```
<BODY>
  <H1> This is first level heading. </H1>
  <H2> This is second level heading. </H2>
  <H3> This is third level heading. </H3>
  <H4> This is fourth level heading. </H4>
  <H5> This is fifth level heading. </H5>
  <H6> This is sixth level heading. </H6>
</BODY>
```

**Output :**

This is first level heading.

This is second level heading.

This is third level heading.

This is fourth level heading.

This is fifth level heading.

This is sixth level heading.

**2. Drawing lines :**

- a. The tag <HR> draws lines and horizontal rules.
- b. This tag draws a horizontal line across the whole page, wherever specified.
- c. The attributes to the <HR> tag are :

**i. ALIGN :**

1. Aligns are the lines on the browser screen, which is by default, aligned to the center of the screen.
2. ALIGN = LEFT will align the line to the left of the screen.
3. ALIGN = RIGHT will align the line to the right of the screen.
4. ALIGN = CENTER will align the line to the center of the screen.

- ii. **SIZE** : Changes the size of the rule.
- iii. **WIDTH** : Sets the width of the rule. It can be set to a number of pixels, or to a percentage of the available width.

**Example :** Welcome to Quantum Publication.

<HR ALIGN = "LEFT" WIDTH = "10" SIZE =

3. **Text styles :** For text styling we use three different tags :
- Bold** : Displays text in BOLDFACE style. The tags <B>-----</B>
  - Italic** : Displays text in ITALIC. The tags used are <I>-----</I>
  - Underline** : Displays text as UNDERLINED. The tags used are <U>-----</U>.

**Example :** <BODY>

<B> Welcome to Quantum Publication. </B>  
<I> Welcome to Quantum Publication. </I>  
<U> Welcome to Quantum Publication. </U>  
</BODY>

**Output :**

Welcome to Quantum Publication.  
Welcome to Quantum Publication.  
Welcome to Quantum Publication.

4. **Center** : <CENTER> ----- </CENTER> tags are used to center everything found between tags, text, lists, images, rules, tables or other page element.

**Example :** <CENTER> Welcome to Quantum Publications! </CENTER>

**Output :** Welcome to Quantum Publications!

5. **Text spacing (Indenting text) :**
- The tag used for inserting blank spaces in an HTML document is <SPACER>.
  - It is supported by Netscape browser only.
  - The attributes used for <SPACER> are :

- TYPE** : To specify whether space has to be left horizontally or vertically.
- SIZE** : Indicates the amount of space to be left. Size accepts any integer.

**Example :** Welcome to Quantum Publications <BR> <SPACER TYPE = "HORIZONTAL" SIZE = 90>  
Knowledge Must Grow!

**Output :** Welcome to Quantum Publications  
Knowledge Must Grow!

#### 6. Font setting tags :

- All text specified within the tags <FONT> and </FONT> will appear in the font, size and colour are specified as attributes of the tag <FONT>.
- The attributes are :
  - FACE** : Sets the font to the specified font name.
  - SIZE** : Sets the size of the text. SIZE can take values between 1 and 7. The default size used is 3.
  - COLOUR** : Sets the colour of the text. COLOUR can be set to an English language colour name or to a hexadecimal triplet.

**Example :** <FONT FACE = "Arial" SIZE = "6" COLOUR = "Black">

Welcome to Quantum Publication. </FONT>

**Output :** Welcome to Quantum Publication.

#### Que 2.4. What are the different types of lists available in HTML ?

Give an example of each type.

#### Answer

Lists are used for giving the hyperlinks in the form of lists in an HTML document. Types of list are :

##### 1. Unordered list :

- An unordered list starts with the tag <UL> and ends with </UL>.
- Each list item starts with the tag <LI>.
- The attributes that can be specified with <LI> : TYPE with three values as FILLROUND, CIRCLE, SQUARE. The value of attribute changes the style of the bullet.
- It is not necessary to use <LI> tag for ending the list.

**Example :** Quantum series is also available for :

<UL TYPE = FILLROUND>

<LI> Computer Science  
<LI> Information Technology  
<LI> MCA  
<LI> Electronics  
<LI> Mechanical  
<LI> Civil  
</UL>

**Output :** Quantum series is also available for :

- Computer Science
- Information Technology
- MCA
- Electronics
- Mechanical
- Civil

### 2. Ordered list :

- i. An ordered list starts with the tag <OL> and ends with </OL>.
- ii. Each list items start with the tag <LI>.
- iii. The attributes that can be specified with <OL> are :
  - a. **TYPE** : It can have five values which are 1, A, a, I, i.
  - b. **START** : It can be set of any numerical value.
  - c. **VALUE** : It is used for changing the numbering sequence in the middle of ordered list.

**Example :** Quantum series for Information technology include:

```
<OL TYPE = "A" START = "3">
<LI> Software Engineering
<LI> Web Technology
<LI> DBMS
<LI> Computer Organization
</OL>
```

**Output :** Quantum series for Information technology include:
 

3. Software Engineering
4. Web Technology
5. DBMS
6. Computer Organization

### 3. Definition list :

- i. Definition list values appear within <DL> and </DL> tags.
- ii. Definition lists consists of two parts :
  - a. Definition Term <DT>
  - b. Definition Description <DD>

**Example :**

```
<DL>
<DT> DBMS
<DD> Database Management System
<DT> CO
<DD> Computer Organization
```

</DL>

**Output :** DBMS

Database Management System

CO

Computer Organization

### Que 2.5. Explain the table tag with its attributes in detail.

#### Answer

All table related tags are included between the <TABLE> - - - - </TABLE> tags.

1. **Table row :** Rows of a table is described between the <TR> - - - </TR> tags. Table rows are of two types :
  - a. **Header rows :** A table header row is defined using <TH> - - - </TH> tags. Header row in a table is that which spans across columns of table and give the information stored in it.
  - b. **Data rows :** Data cells placed in the horizontal plane creates a data row. There could be single or multiple data cells. Data cells are the columns in a table.
2. **Table data :** Table data tags used for displaying data in table data cells using <TD> - - - </TD> tags. These tags must be needed inside the <TR> - - - </TR> tags.
3. **Table caption :**
  - a. Often tables need to be given a heading, which gives the reader a context for the information in the tables.
  - b. Table headings are called captions. Captions are given to the table by using the <CAPTION> - - - </CAPTION> tags.
  - c. This tag has attribute ALIGN with two values TOP and BOTTOM.
4. The attributes that can be included in the <TABLE> tag are :
  - a. **ALIGN :** Horizontal alignment is controlled by the ALIGN attribute. It can be set to LEFT, CENTER or RIGHT.
  - b. **VALIGN :** Controls the vertical alignment of cell contents. It accepts the values TOP, MIDDLE or BOTTOM.
  - c. **WIDTH :** Sets the WIDTH to a specific number of pixels or to a percentage of the available screen width. If width is not specified, the data cell is adjusted based on the cell data value.
  - d. **BORDER :** Controls the border to be placed around the table. The border thickness is specified in pixels.
  - e. **CELL PADDING :** Controls the distance between the data in a cell and the boundaries of the cell.

2-10 Z (CS-5/IT-6)

### Web Page Design

- f. **CELL SPACING :** Controls the spacing between adjacent cells.
- g. **COLSPAN :** Width of the cell in terms of number of columns used when a cell occupies more than one column.
- h. **ROWSPAN :** Height of the cell in terms of rows are used when a cell occupies more than one row.

**Que 2.6.** How do you make an image clickable in HTML? Illustrate with an example.

AKTU 2014-15, Marks

Explain image in HTML.

OR

### Answer

1. HTML accepts two picture file formats : .gif and .jpg (.jpeg).
2. To add an image to web page we used the <IMG> tag, which takes name of the image file as an attribute, also control the height, width, border etc.
3. The <IMG> tag takes the following attributes :
  - a. **ALIGN :** The ALIGN attribute allows us to position an image relative to the line of text. All graphical web browsers recognize these values TOP, MIDDLE and BOTTOM.
  - b. **BORDER :** Specifies the size of the border to place around the image.
  - c. **WIDTH :** Specifies the width of the image in pixels.
  - d. **HEIGHT :** Specifies the height of the image in pixels.
  - e. **HSPACE :** Indicates the amount of space to the left and right of the image.
  - f. **VSPACE :** Indicates the amount of space to the top and bottom of the image.
  - g. **ALT :** Indicates the text to be displayed in case the browser is unable to display the image specified in the SRC attribute.
  - h. **SRC :** Specifies the location and name of the image file.

### Creating an image link :

We can also make a clickable link (image displayed with a blue border) indicating that it is a hyperlink.

**Example :** <A HREF = "QUANTUM.HTM">

<IMG SRC = "QUANTUM.GIF" BORDER = "10"> </A>

This tells the web browser that the image file "QUANTUM.GIF" is clickable and any click on the image should be directed to the home page (QUANTUM.HTM).

**Que 2.7.** Explain frames in HTML with example.

### Web Technology

2-11 Z (CS-5/IT-6)

### Answer

1. The HTML frame is a powerful feature that enables a web page to be broken into different unique sections, that although related and operate independent of each other.
2. Each 'frame' can be loaded with a different document and hence, allow multiple HTML documents to be seen concurrently.
3. The HTML tags that divide a browser screen into two or more HTML recognizable unique regions is the <FRAMESET> ... </FRAMESET> tags. Each unique region is called a 'frame'.

### <FRAMESET> tag :

1. The splitting of a browser screen into frames is accomplished with the <FRAMESET> and </FRAMESET> tags embedded into the HTML document.
2. The <FRAMESET> ... </FRAMESET> tags require one of the following two attributes depending on whether the screen has to be divided into rows or columns.
  - a. **Rows :**
    - i. This attribute is used to divide the screen into multiple rows.
    - ii. It can be set equal to a list of values, depending on the required size of each row.
    - iii. The values can be number of pixels, expressed as a percentage of the screen resolution and the symbol \*, which indicates the remaining space.
  - b. **Cols :**
    - i. This attribute is used to divide the screen into multiple columns.
    - ii. It can be set equal to a list of values, depending on the required size of each column.
    - iii. The values can be numbers of pixels, expressed as a percentage of the screen resolution and the symbol \*, which indicates the remaining space.

**Example :** <FRAMESET Rows = "33%, 33%, 33%">

: Divides the browser screen into three equal horizontal sections.

<FRAMESET Cols = "50%, 50%">

: Splits the 1<sup>st</sup> horizontal section into two equal vertical sections.

</FRAMESET>

<FRAMESET Cols = "50%, 50%">

: Splits the 2<sup>nd</sup> horizontal section into two equal vertical sections.  
<FRAMESET>  
</FRAMESET>

**<FRAME> tag :**

- Once the browser screen is divided into rows and columns, each unique section defined can be loaded with different HTML documents.
- This is achieved by using the <FRAME> tag, which takes the following attributes :
  - SRC = "url"** : Indicates the URL of the document to be loaded in the frame.
  - MarginHeight = "n"** : Specifies the amount of white space to left at the top and bottom of the frame.
  - MarginWidth = "n"** : Specifies the amount of white space to left along the sides of the frame.
  - Name = "name"** : Gives the frame a unique name so it can be targeted by other documents. The name given must begin with alphanumeric character.
  - Noresize** : Disables the frames resizing capability.
  - Scrolling** : Controls the appearance of horizontal and vertical scrollbars in a frame. This takes the values YES/NO/AUTO.

**Example :**

```
<HTML>
<FRAMESET Rows = "30%, **">
: Divides the screen into 2 rows, one occupying 30% of the
screen, and other occupying the remaining space.
<FRAMESET Cols = "50%, 50%">
: Divide the first row into 2 equal columns, each 50% of the
screen.
<FRAME Src = "File1.html">
: Loads the 1st frame with File1.html
<FRAME Src = "File2.html">
: Loads the 2nd frame with File2.html
</FRAMESET>
<FRAMESET Cols = "50%, 50%">
: Divides the second row into 2 equal columns, 50% of the
screen.
<FRAME Src = "File3.html">
: Loads the 1st frame with File3.html.
```

<FRAME Src = "File4.html">

: Loads the 2<sup>nd</sup> frame with File4.html.

</FRAMESET>

</FRAMESET>

</HTML>

- Que 2.8.** Write HTML code to develop a web page having two frames that divide the page into two equal rows and divides the first row into equal columns. Fill each with the different background colour.

**AKTU 2013-14, Marks 06**

**Answer**

```
<html>
<body>
<frame Rows = "50%, 50%">
<frame Cols = "50%, 50%">
<frame SRC = "File 1.html"> </frame>
<frame SRC = "File 2.html"> </frame>
</frame>
<frame SRC = "File 3.html"> </frame>
</frame>
</body>
</html>
```

**For background colour :****For File 1.html :**

```
= <html>
<body BGCOLOR = "RED">
</body>
</html>
```

**For File 2.html :**

```
= <html>
<body BGCOLOR = "GREEN">
</body>
</html>
```

**For File 3.html :**

```
= <html>
<body BGCOLOR = "BLUE">
</body>
</html>
```

## Web Page Designing

2-14 Z (CS-5/IT-6)

**Que 2.9.** Discuss forms in HTML. Explain various input items used in HTML forms.

### Answer

1. All the input elements should be enclosed within the opening <FORM> and closing </FORM> tags like this :  
<FORM> The input elements go here </FORM>
2. The following are the important attributes for the form tag :
  - a. **ACTION :**
    - i. This attribute specifies the page which handles the input from the user.
    - ii. Usually, this will be a script or a CGI program which processes the data supplied by the user.
  - b. **METHOD :**
    - i. In simple terms, if we use GET method, the input values are passed as part of the URL.
    - ii. If it is POST, the information is sent to the server as part of the data body and will not be visible in the URL box in the user's browser. If we do not specify the method, GET is taken by default.

### Various input items in HTML forms are :

1. **Text input :**
  - a. Text input is used to collect single line of text from the user like name, e-mail address etc.
  - b. It is the most commonly used input type.
  - c. A text input item can be defined like this :  
`<INPUT TYPE="TEXT" NAME="FirstName">`
2. **Submit button :**
  - a. After entering the data, the user presses the submit button which triggers the browser to send the data to the server.
  - b. We can add a submit button to the form using the 'submit' input type.
  - c. We can add a submit button to our HTML form using the following code :  
`<INPUT TYPE="SUBMIT" NAME="name" VALUE="label">`
3. **Checkbox :**
  - a. Checkbox is used to select or deselect the multiple option from a set of options.
  - b. If we specify CHECKED, the checkbox will be checked by default.

## Web Technology

2-15 Z (CS-5/IT-6)

### 4. Radio button :

- a. Radio buttons are used for selecting one item from multiple available choices.
- b. When the user selects a button in the set, all other buttons in the set are deselected.
- c. The individual button in a set is created using input type "RADIO".

### 5. Select lists :

- a. <SELECT> tag is used to create a list of items for the user to select from.
- b. Different from the input tags the list has two HTML tags associated with the <SELECT> tag and the <OPTION> tag. We can create a list using the <SELECT> </SELECT> tag and the items in the list using the <OPTION> tag.

### 6. Password input :

- a. Login screens usually have a password field where the user enters his password. We can create a password field by using the input type PASSWORD.
- b. A password field can be created using the following code :  
`<INPUT TYPE="PASSWORD" NAME="pwd">`

### 7. Uploading a file :

- a. Some HTML form allows the user to upload a file.
- b. The input type FILE lets the user to upload a file to the server.
- c. Here is the syntax of FILE input type :  
`<INPUT TYPE="FILE" NAME="name" VALUE="filename">`

### 8. Button input :

- a. A button appears in the form. We must specify JavaScript code as the value of the ONCLICK attribute to determine what happens when the user clicks the button.
- b. We cannot use button without the client-side scripting language.
- c. An example of using the button input type is given as :  
`<INPUT TYPE="BUTTON" OnClick="javascript:alert('Clicked')"; VALUE="Click!">`

This code displays a button with label "Click!". On pressing the button a message box with message 'Clicked' is displayed.

### 9. Reset the form :

- a. The RESET input types can be used to reset the form.
- b. When the user presses the reset button, all the elements in the form are reset to their default values.

## Web Page Designing

2-16 Z (CS-5/IT-6)

**Que 2.10.** Write a HTML code to design a 'Student Registration Form'. (Make Assumptions).

AKTU 2014-15, Marks 06

### Answer

```
<html>
<head>
<script type = "text/JavaScript" src = "validate.js"></script>
</head>
<body>
<form action = "#" name = "StudentRegistration"
onsubmit = "return(validate());">
<table cellpadding = "2" width = "20%" bgcolor = "99FFFF"
align = "center" cellspacing = "2">
<tr>
<td colspan = "2">
<center><font size = "4"><b>Student Registration Form</b></font></center>
</td>
</tr>
<tr>
<td>Name</td>
<td><input type = "text" name = "textnames" id = "textname"
size = "30"></td>
</tr>
<tr>
<td>Father Name</td>
<td><input type = "text" name = "fathername" id = "fathername" size =
"30"></td>
</tr>
<tr>
<td>Personal Address</td>
<td><input type = "text" name = "personaladdress" id =
"personaladdress" size = "30"></td>
</tr>
<tr>
<td>Sex</td>
```

## Web Technology

2-17 Z (CS-5/IT-6)

```
<td><input type = "radio" name = "sex" value = "male"
size = "10">Male
<input type = "radio" name = "sex" value = "Female"
size = "10">Female</td>
<input type = "radio" name = "sex" value = "Female"
size = "10">Female</td>
</tr>
<tr>
<td>Course</td>
<td><select name = "Course">
<option value = "-1" selected>Select</option>
<option value = "BTech">B.TECH</option>
<option value = "MCA">MCA</option>
</select></td>
</tr>
<tr>
<td>E-mail</td>
<td><input type = "text" name = "e-mail" id = "e-mail"
size = "30"></td>
</tr>
<tr>
<td>DOB</td>
<td><input type = "text" name = "dob" id = "dob" size = "30"></td>
</tr>
<tr>
<td>Mobile No. </td>
<td><input type = "text" name = "mobileno" id = "mobileno."
size = "30"></td>
</tr>
<tr>
<td colspan = "2"><input type = "submit" value = "Submit Form"> </td>
</tr>
</table>
</form>
</body>
</html>
```

**Output :**

2-18 Z (CS-5/IT-6)

Web Page Design

**Student Registration Form**

Name	<input type="text"/>
Father Name	<input type="text"/>
Personal Address	<input type="text"/>
Sex	<input type="radio"/> Male <input type="radio"/> Female
Course	<input type="text"/> Select ▼
E-mail	<input type="text"/>
DOB	<input type="text"/>
Mobile No.	<input type="text"/>
<input type="button" value="Reset"/>	<input type="button" value="Submit Form"/>

Que 2.11. Design a HTML form for a railway reservation system

**Answer**

AKTU 2016-17, Marks 10

```

<html>
<head>
<script type = "text/JavaScript" src = "validate.js"></script>
<body>
<form action = "#" name = "RailwayReservationSystem"
onsubmit = "return(validate());">
<table cellpadding = "2" width = "20%" bgcolor = "99FFFF"
align = "center" cellspacing = "2">
<tr>
<td colspan = "2">
<center><font size = "4"><b>Railway Reservation System</b></font>
</center>
</td>
<tr>
<td>From</td>
<td><input type = "text" name = "from" id = "from"
size = "30"></td>
</tr>
<tr>
<td>To</td>
<td><input type = "text" name = "to" id = "to" size = "30"></td>

```

Web Technology

2-19 Z (CS-5/IT-6)

```

</tr>
<tr>
<td>Train No/Name</td>
<td><input type = "text" name = "trainno"
id = "trainno" size = "30"></td>
</tr>
<tr>
<td>Class</td>
<td><select name = "Class">
<option value = "-1" selected>Select</option>
<option value = "SL">SL</option>
<option value = "3A">3A</option>
<option value = "2A">2A</option>
<option value = "1A">1A</option>
</select></td>
</tr>
<tr>
<td>No. of seats</td>
<td><Select name = "No. of seats">
<option value = "-1" selected>Select</option>
<option value = "1">1</option>
<option value = "2">2</option>
<option value = "3">3</option>
</select></td>
</tr>
<tr>
<td>Passenger</td>
<td><input type = "text" name = "age" size = "2"></td>
<td><Select name = "gender">
<option value = "M">M</option>
<option value = "F">F</option>
</select></td>
</tr>
<tr>
<td><input type = "text" name = "P1" size = "30"></td>
<td><input type = "text" name = "age" size = "2"></td>
<td><Select name = "gender">

```

```

<option value = "M">M</option>
<option value = "F">F</option></select>
</td></tr>
<tr>
<td><input type = "text" name = "P3" size = "30"></td>
<td><input type = "text" name = "age" size = "2"></td>
<td><Select name = "gender">
<option value = "M">M</option>
<option value = "F">F</option></select>
</td></tr>
<tr>
<td>Address</td>
<td><input type = "text" name = "address" id = "address" size = "50"></td>
</tr>
<tr>
<td>Payment Mode</td>
<td><input type = "radio" name = "Paymentmode" value = "Credit/Debit Card">Credit/Debit Card
<input type = "radio" name = "Paymentmode" value = "Wallet/UPI">UPI
<input type = "radio" name = "Paymentmode" value = "netbanking">Net Banking</td>
</tr>
<tr>
<td>Mobile No. </td>
<td><input type = "text" name = "mobileno" id = "mobileno." size = "30"></td>
</tr>
<tr>
<td><input type = "reset"></td>
<td colspan = "2"><input type = "submit" value = "Submit Form"></td>
<input type = "Cancel" value = "Cancel">
</td>
</tr>
</table>
</form>
</body>
</html>

```

**Output :**

**Que 2.12.** What do you mean by Cascading Style sheet (CSS) ?  
What are the advantages and features of CSS ?

**Answer**

1. Cascading Style Sheet or CSS enables us to separate the content of HTML documents from the presentation.
2. A single file or a small group of files could define the presentation format for the entire website. Thus, any format or presentation changes across the website would be controlled through these CSS files.
3. To define styles, we use the `<style>` element.
4. To define properties for the documents, we specify the attributes for the document tags within the `<style>`.
5. When defining the style for a template HTML file, the style element is placed within the document `<head>` and not in the `<body>`.

**Advantages of CSS :**

1. To make the web page attractive.
2. Faster download of web page.
3. Increase visual appearance of web page.

**Various features of CSS are :**

- 1. Cascading :**
  - a. This is the capability provided by CSS to allow style information from several sources to be blended together.
  - b. The cascade defines an ordered sequence of style sheets where rules in later sheets have greater precedence than earlier ones.
  - c. By storing these separately, style sheets can be reused.
- 2. Flexible placement of style information :**
  - a. Placing style sheets in separate files makes them easy to reuse.
  - b. To make it easier to manage style on a site basis, this specification describes how to use HTTP headers to set the style sheets to be applied to a document.
- 3. Independence from specific style sheet languages :**
  - a. This allows for a range of such languages to be used, for instance simple ones for the majority of users and much more complex ones for the minority of users with highly specialized needs.
- 4. Media dependencies :**
  - a. HTML allows authors to specify documents in a media-independent way.
  - b. This allows users to access web pages using a wide variety of devices and media.
  - c. This allows user agents to avoid retrieving in appropriate style sheets.

**Que 2.13.** What do you understand by CSS ? Explain different types of CSS with examples.

AKTU 2013-14, Marks 06

What is CSS ? What are different ways to create them ? Explain with example.

AKTU 2017-18, Marks 10

**Answer**

CSS : Refer Q. 2.12, Page 2-21Z, Unit-2.  
Different types of CSS :

- 1. Inline CSS :**
  - a. An inline CSS is used to apply a unique style to a single HTML element.
  - b. An inline CSS uses the style attribute of an HTML element.

For example : Following example sets the text color of the `<h1>` element to blue :

```
<h1 style="color:blue;">This is a Blue Heading</h1>
```

**2. Internal CSS :**

- a. An internal CSS is used to define a style for a single HTML page.
- b. An internal CSS is defined in the `<head>` section of an HTML page, within a `<style>` element.

For example :

```
<!DOCTYPE html>
<html>
<head>
<style>
body {background-color : powderblue;}
h1 {color : blue;}
p {color : red;}
</style>
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

**3. External CSS :**

- a. An external style sheet is used to define the style for many HTML pages.
- b. An external style sheet is used to change the look of an entire website.
- c. To use an external style sheet we add a link in the `<head>` section of the HTML page.

For example :

```
<!DOCTYPE html>
<html>
<head>
<link rel="stylesheet" href="styles.css">
</head>
<body>
<h1>This is a heading</h1>
<p>This is a paragraph.</p>
</body>
</html>
```

**2-24 Z (CS-5/IT-6)**

**Que 2.14.** Explain the properties and uses of CSS.

**Answer**

Properties of CSS are :

1. **Font properties :**

- a. **Font-family :** Denotes font of the text.
- b. **Font-size :** Denotes the size of the text.
- c. **Font-style :** Denotes the style of the text i.e., normal, bold etc.
- d. **Font-weight :** Denotes the weight or darkness of the font.

2. **Text properties :**

- a. **Word-spacing :** Denotes the space between words.
- b. **Vertical-align :** Denotes the vertical positioning of the images, with respect to the baseline.
- c. **Text-align :** Specifies the alignment of the text. The possible values are center, justify etc.
- d. **Text-transform :** Denotes the transformation of text. The values are capitalize uppercase, lowercase etc.
- e. **Text-decorate :** Denotes the text decoration. The standards for this property include blink, line-through, overline, underline etc.

3. **Colour and background properties :**

- a. **Colour :** Used to set the colour of the text.
- b. **Background-colour :** This property set an element background colour.
- c. **Background-image :** Associates a background image with an element.
- d. **Background-position :** Specifies how a background image is positioned.

4. **Box properties :**

- a. **Margin-properties :** The individual margins for a block element can be set using margin-top, margin-right, margin-bottom, margin-left.
- b. **Border properties :**
  - i. **Border-style**
  - ii. **Border-width**
  - iii. **Border-colour**

Uses of CSS :

1. CSS is used in the Web document for presentation purpose.

**Web Technology**

**2-25 Z (CS-5/IT-6)**

2. It is used to separate the Web content from the Web presentation.
3. It is used to enhance the features of Web pages like formatting styles.
4. CSS helps in faster downloading of the pages.

**PART-2**

**Document Type Definition, XML : DTD, XML Schemas, Object Models, Presenting and Using XML, Using XML Processors : DOM and SAX, Dynamic HTML.**

**CONCEPT OUTLINE : PART-2**

- DTD defines the documents structure with a list of various elements and attributes.
- DOM is an object model for representing HTML, XML and related formats.
- XML schema is an XML based alternative to DTD.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 2.15.** Discuss various types of DTDs (Document Type Definition) in XML. Which type of DTD is preferable and why ?

**AKTU 2014-15, Marks 06**

**OR**

**What is DTD ? Explain various types of DTDs.**

**Answer**

1. A Document Type Definition (DTD) defines the basic building blocks of an XML document.
2. It defines the document structure with a list of various elements and attributes.
3. A DTD can be declared inline inside an XML document, or as an external reference.

**Types of DTD :**

1. **Internal DTD declaration :**

If the DTD is declared inside the XML file, it should be wrapped in a DOCTYPE definition with the following syntax:

<!DOCTYPE >

Scanned by CamScanner

## Web Page Designing

2-26 Z (CS-5/IT-6)

- 2. External DTD declaration :** If the DTD is declared in an external file, it should be wrapped in a DOCTYPE definition with the following syntax :  
**<!DOCTYPE root-element SYSTEM "filename">**
- External DTD is preferable because :** They can be used in more than one document. So, it is easy to change in one external DTD rather than changing in all internal DTD files.

**Que 2.16.** What is DTD ? What are the differences between external and internal DTD ? Use suitable example.

**AKTU 2015-16, Marks 10**

**Answer** DTD : Refer Q. 2.15, Page 2-25Z, Unit-2.

S.No.	External DTD	Internal DTD
1.	In external DTD, elements are declared outside the XML files.	In internal DTD, elements are declared within the XML files.
2.	The syntax for external DTD is : <b>&lt;!DOCTYPE root-element SYSTEM "file-name"&gt;</b> where file-name is the file with .dtd extension.	The syntax for internal DTD is : <b>&lt;!DOCTYPE root-element [element-declarations]&gt;</b> where root-element is the name of root element and element-declarations is where we declare the elements.
3.	To reference it as external DTD, standalone attribute in the XML declaration must be set as no. This means, declaration includes information from the external source.	To reference it as internal DTD, standalone attribute in XML declaration must be set to yes. This means the declaration works independent of external source.

**Example of internal DTD :**

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<!DOCTYPE address [
    <!ELEMENT address (name,company,phone)>
    <!ELEMENT name (#PCDATA)>
    <!ELEMENT company (#PCDATA)>
]
```

Web Technology

2-27 Z (CS-5/IT-6)

**<!ELEMENT phone (#PCDATA)> ]>**

**<address>**

**<name>Pratibha Patil</name>**

**<company>Quantum</company>**

**<phone>(011) 123-4567</phone>**

**</address>**

**<address>Pratibha Patil</name>**

**<company>Quantum</company>**

**<phone>(011) 123-4567</phone>**

**</address>**

The content of the DTD file address.dtd are as shown :

**<!ELEMENT address (name,company,phone)>**

**<!ELEMENT name (#PCDATA)>**

**<!ELEMENT company (#PCDATA)>**

**<!ELEMENT phone (#PCDATA)>**

**Que 2.17.** Explain the role of DTD in XML and also describe its types with an example.

**AKTU 2017-18, Marks 05**

**Answer**

**Role of DTD :**

1. The role of a DTD is to define the legal building blocks of an XML document.
2. It defines the document structure with a list of legal elements.
3. A DTD can be declared inline in our XML document, or as an external reference.

**Types of DTD :** Refer Q. 2.15, Page 2-25Z, Unit-2.

**Que 2.18.** What is XML ? Discuss the significance of XML. How is XML different from HTML ? Explain the process of XML parsing.

**How are they useful ?**

**AKTU 2015-16, Marks 10**

**AKTU 2016-17, Marks 15**

## Web Page Designing

2-28 Z (CS-5/IT-6)

### Answer

1. XML is a markup language for documents containing structured information which contains both content and some indication of the role of content.
2. Extensible Markup Language, abbreviated as XML, describes a class of data objects called XML documents and partially describes the behaviour of computer programs which process them.
3. XML documents are made up of storage units called entities, which contain either parsed or unparsed data.

### Significance of XML:

1. XML can store and organize just about any kind of information in a form according to our needs.
2. With its clear, simple syntax and unambiguous structure, XML is easy to read and parse.
3. XML is easily combined with stylesheets to create formatted documents in any style.

### Difference between XML and HTML:

S.No.	XML	HTML
1.	XML is designed to describe data and to focus on what data is about.	HTML is designed to display data and to focus on how data looks like.
2.	XML is about describing information.	HTML is about displaying information.
3.	XML tags are not predefined.	HTML tags are predefined.
4.	In XML, data is stored in separate XML file.	In HTML, data is stored inside the HTML tags.

### Process of XML parsing :

```
<?xml version = "1.0" encoding ="ISO-8859-1"?>
<bookstore>
<book category="cooking">
<title lang="en">Khana Khazana</title>
<author>Sanjeev Kapoor</author>
<year>2015</year>
<price>30.00</price>
</book>
<book category="children">
<title lang="en">Harry Potter</title>
<author>J K. Rowling</author>
<year>2015</year>
```

## Web Technology

2-29 Z (CS-5/IT-6)

price>29.99</price>















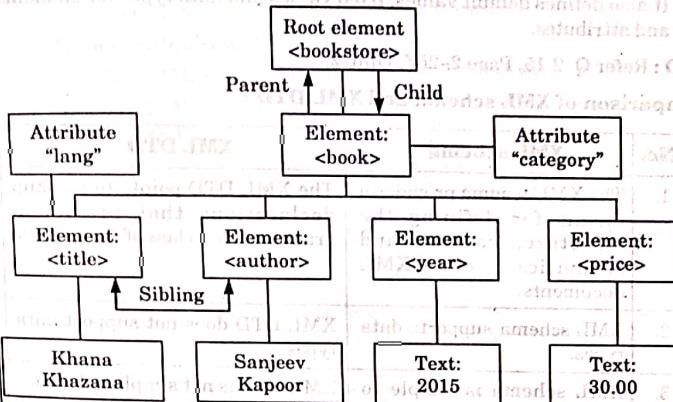


Fig. 2.18.1

### ML parser are useful in :

- Loading the elements of XML document.
- Accessing the elements of XML document.
- Deleting the elements of XML document.
- Changing the elements of XML document.

Ques 2.19. What is XML schema ? Compare XML schema and XML DTD.

OR

What is DTD ? Also explain its differences with XML schema.

AKTU 2017-18, Marks 10

### Answer

XML schema is an XML based alternative to Document Type Definition (DTD).

The goal or purpose of XML schema is to define the building blocks of an XML document.

3. XML schema syntax is well-formed XML, making it possible to use XML tools to edit it.
4. The XML schema language is called as XML Schema Definition (XSD) language.
5. XML schema defines elements, attributes, child elements, order of elements, number of child elements and whether an element is empty or can include text.
6. It also defines default values, fixed values and data types for elements and attributes.

**DTD :** Refer Q. 2.15, Page 2-25Z, Unit-2.

#### Comparison of XML schema and XML DTD :

S.No.	XML schema	XML DTD
1.	The XML schema provides a means for defining the structure, content and semantics of XML documents.	The XML DTD points to mark declarations that provide grammar for a class of document.
2.	XML schema supports data types.	XML DTD does not support data types.
3.	XML schema is simple to learn.	XML DTD is not simple to learn.
4.	It provides more control on XML structure.	DTD provides less control on XML structure.
5.	It uses an XML-based syntax.	DTD uses a unique syntax.
6.	<b>Example :</b> <XS : element name="note"> <XS : ComplexType> <XS: sequence > <XS: element name "to" type="XS: String"/> <XS: element name "from" type="XS: string"/> <XS:element name="heading" type="XS:string"/>	<b>Example :</b> <!DOCTYPE note [ <!ELEMENT note (to, from, heading, body)> <!Element to (#PCDATA)> <!Element from (#PCDATA)> <! Element heading (?)> <!Element body (#PCDATA)>

**Que 2.20.** Give advantages and disadvantages of XML schema.

#### Answer

##### Advantages of XML schema :

1. The schemas are more specific and provide the support for data types.
2. The schema is aware of namespaces.
3. The XML schema is written in XML itself and has a large number of built-in and derived data types.
4. The XML schema is the W3C recommendation. Hence, it is supported by various XML validator and XML processors.

##### Disadvantages of XML schema :

1. The XML schema is complex to design and hard to learn.
2. The XML document cannot be possible if the corresponding schema file is absent.
3. Maintaining the schema for large and complex operations sometimes slows down the processing of XML document.

**Que 2.21.** Define HTML DOM.

**AKTU 2017-18, Marks 05**

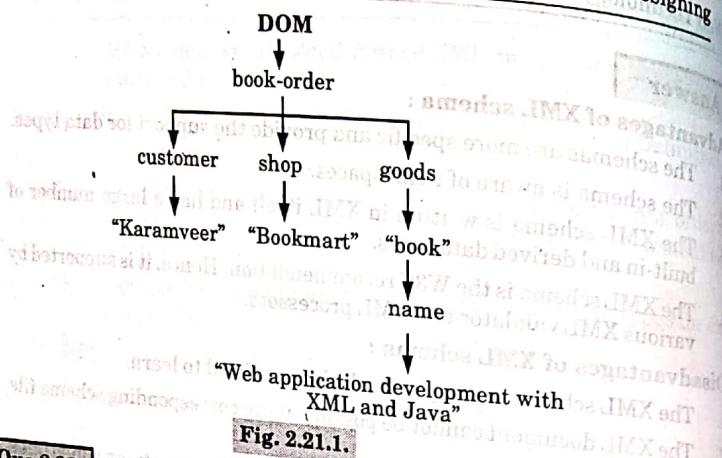
#### Answer

1. The Document Object Model (DOM) is a platform-independent and language-independent standard object model for representing HTML or XML and related formats.
2. The DOM is a W3C (World Wide Web consortium) standard.
3. The DOM defines a standard for accessing documents like XML and HTML.
4. It allows programs and scripts to dynamically access and update the content, structure, and style of document.

**Example :**  
<? xml version = "1.0"?>  
<book-order>  
  <customer> Karamveer </customer>  
  <shop> Bookmart </shop>  
  <goods>  
    <book> "Introduction to Java programming" </book>  
    <name> "Web application development with XML and Java" </name>  
    </book>  
  </goods>  
</book-order>

2-32 Z (CS-5/IT-6)

## Web Page Designing



Que 2.22. Explain COM and DCOM in detail.

OR

AKTU 2016-17, Marks 10

Write a short note on COM/DCOM.

AKTU 2015-16, Marks 05

Answer

COM:

1. The Component Object Model (COM) is a software architecture that allows applications to be built from binary software components.
2. COM is the underlying architecture that forms the foundation for higher-level software services.
3. It is used to enable inter-process communication and dynamic object creation in a large range of programming languages.
4. In a component based system, components interact with each other by calling methods and passing data.
5. COM ensures that there is a standard method of interaction between the components.
6. All the COM objects need to follow these standards when providing functionality.

DCOM:

1. DCOM (Distributed Component Object Model) is a set of Microsoft concepts and program interfaces in which client program objects can request services from server program objects on other computers in a network.
2. DCOM is based on the Component Object Model (COM), which provides a set of interfaces allowing clients and servers to communicate within the same computer.
3. DCOM (Distributed Component Object Model) is a model as COM but it is specially designed for distributed application.

Web Technology Td97

2-33 Z (CS-5/IT-6)

DCOM, which originally was called "Network OLE" extends Microsoft's COM, and provides the communication substrate under Microsoft's COM+ application server infrastructure. DCOM was a major competitor to CORBA.

Que 2.23. What is difference between COM and DCOM?

AKTU 2013-14, Marks 03

Answer

No.	COM	DCOM
1.	COM stands for Component Object Model.	DCOM stands for Distributed Component Object Model.
2.	Component Object Model is executed on client-side environment.	Distributed Component Object Model is executed on server-side environment.
3.	COM objects require installation on the machine from where it is being used.	DCOM requires installation somewhere on the same network.
4.	COM allows reuse of objects.	DCOM do not allow reuse of objects.

Que 2.24. What are XML processors?

Answer

An XML processor is a software module that is used to read XML documents and provide application programs with access to their content and structure.

XML processors are written in Java. Some are validating processors, while others are non-validating.

When reading an XML document, a validating processor checks the validity constraints and the well-formed constraints defined in XML 1.0 recommendation.

A validating XML processor is one of the most robust and faithful implementation of XML processor.

The validating processor XML for Java is a Java class library.

Que 2.25. Explain the term SAX with suitable example.

Answer

SAX stands for Simple API for XML and works directly with an XML.

2-34 Z (CS-5/IT-6)

2. SAX is an event-driven API that allows us to interpret a web file that uses XML.
3. SAX takes the control of event specifies by the programmer and handles the situation.
4. SAX is a simpler interface than DOM and is appropriate where many or very large files are to be processed.
5. It also helps in manipulating the data content.
6. SAX is faster and uses less memory than DOM.

**Example :**

```
<?xml version = "1.0"?>
<book-order>
  <customer> Karamveer </customer>
  <shop> Bookmart </shop>
  <goods>
    <book>
      <name> Web application development with XML and Java </name>
    </book>
  </goods>
</book-order>
```

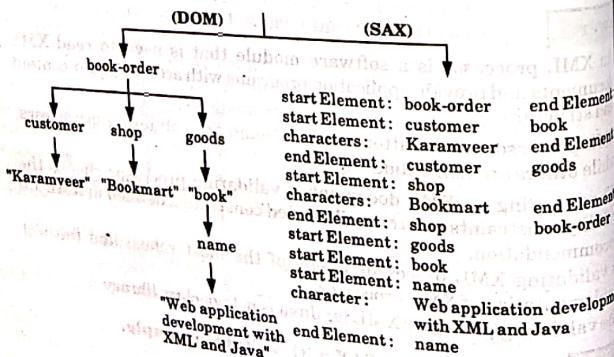


Fig. 2.25.1.

**Que 2.26. Explain XML processing with SAX.****Answer**

1. SAX (Simple API for XML) is a serial access parser API for XML.
2. SAX provides a mechanism for reading data from an XML document.
3. A parser which implements SAX (i.e. a SAX parser) functions as a stream parser with an event-driven API.
4. The user defines a number of callback methods that will be called when events occur during parsing.
5. The SAX events include :
  - a. XML Text nodes
  - b. XML Element nodes
  - c. XML Processing Instructions
  - d. XML Comments
6. Events are triggered when each of these XML features are encountered and again when the end of them is encountered.
7. XML attributes are provided as part of the data passed to element events.
8. SAX parsing is unidirectional i.e., previously parsed data cannot be re-read without starting the parsing operation again.

**Que 2.27. What is DHTML ? Write difference between HTML and DHTML.****Answer**

1. DHTML stands for Dynamic Hyper Text Markup Language.
2. DHTML is a combination of technologies used to create dynamic web pages.
3. DHTML means a combination of HTML, CSS, DOM and JavaScript.
4. DHTML is designed to enhance the user experience on web.
5. DHTML includes the following features :
  - a. Dynamic content, which allows the user to dynamically change web page content.
  - b. Dynamic positioning of web page elements.
  - c. Dynamic style, which allows the user to change the web page's colour, font, size or content.

**Difference :**

S. No.	HTML	DHTML
1.	HTML is used to create static web pages.	DHTML is used to create dynamic web pages.
2.	HTML consists of simple HTML tags.	DHTML is made up of HTML tags + Cascading Style Sheets (CSS) + JavaScripts.
3.	HTML does not allow to alter the text and graphics on the web page unless web page gets changed.	DHTML allows to alter the text and graphics of the web page without changing the entire web page.
4.	HTML web pages are simple but less interactive.	DHTML web pages are complex but more interactive.

**PART-1**

*Scripting : JavaScript : Introduction, Documents, Forms, Statements, Functions, Objects : Introduction to AJAX.*

**CONCEPT OUTLINE : PART-1**

- JavaScript is a scripting language that reflects the object orientation of web pages.
- Loop statements are used when we want to run JavaScript multiple times.
- There are various JavaScript objects like string object, math object, date object, array object.
- JavaScript can be used to validate input data in HTML forms.
- AJAX stands for Asynchronous JavaScript and XML.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 3.1.** What is JavaScript ? How it works ? What are the features of JavaScript ?

**OR**

Discuss in detail about JavaScript. Write a JavaScript program to find largest among 5 numbers.

**Answer**

1. JavaScript is a scripting language which enables web authors to design interactive sites.
2. JavaScript can interact with HTML source code, enabling web authors to spice up their sites with dynamic content.
3. JavaScript is an open source language that anyone can use without purchasing a license.

**Working of JavaScript :**

1. When the browser loads a web page, the HTML parser creates the DOM.
2. Whenever parser encounters JavaScript directive, it is handed over the JavaScript engine and loads the external and inline code.

### 3-3 Z (CS-5/IT-6)

#### Web Technology

3. After HTML and CSS parsing is completed, JavaScript is executed in order they were found in web page and DOM is updated and rendered by the browser.

#### JavaScript has several features :

1. **Programming tool:** JavaScript is a scripting language with very simple syntax.
2. **Can produce dynamic text into an HTML page :** For instance, the JavaScript statement - `document.write("<h1>" + name + "</h1>");` results into the HTML output `<h1> Atul </h1>`, if the variable name contains the text Atul.
3. **Reaching to events :** JavaScript code executes when something happens such as when page has finished loading, when a user clicks on an HTML element.
4. **Reading and writing HTML elements :** JavaScript can read and change the content of an HTML element.
5. **Validate data :** JavaScript can be used to validate form data before it is submitted to a server, and thus saves the server from extra processing.

#### Program to find largest among 5 numbers :

```

<HTML>
<HEAD>
<TITLE> Largest among five numbers </TITLE>
</HEAD>
<BODY>
<H1> Largest </H1>
<p>
<SCRIPT Type = "text/JavaScript">
var i, n;
var max=0;
n=parseInt(prompt("Enter array size"));
var a=new Array(5);
for(i=0;i<4;i++)
{
a[i]=parseInt(prompt("Enter numbers"));
}
//finding Maximum//
max=a[0];
for(i=1;i<4;i++)
{
if(max<a[i])
max=a[i];
}
document.write("Array Elements : - ");
for(i=0;i<n;i++)
{
  
```

### 3-4 Z (CS-5/IT-6)

#### Scripting and Networking

```

document.write(a[i]+");
}
document.write("<br>"+"Maximum=" +max+"<br>");
</SCRIPT>
</BODY>
</HTML>
  
```

**Que 3.2.** What is the difference between Java and JavaScript ? Describe the strengths and weakness of JavaScript.

**AKTU 2013-14, Marks 06**

OR  
Compare and contrast Java and JavaScript.

**AKTU 2016-17, Marks 10**

#### Answer

#### Difference between Java and JavaScript :

S. No.	Java	JavaScript
1.	Java is an object-oriented programming language.	JavaScript is an object based scripting language.
2.	Java is strongly typed language and type checking.	JavaScript is very flexible in data type.
3.	Objects in Java are static.	Objects in JavaScript are dynamic.
4.	It can be used to create standalone application.	It cannot be used to create standalone application.
5.	Variables in Java are declared as : int num.	Variables in JavaScript are declared as : var myname.

#### Strengths/Advantages of JavaScript :

1. **An interpreted language :** JavaScript is an interpreted language, which requires no compilation steps.
2. **Quick development :** JavaScript does not require time consuming compilations, scripts can be developed in a short period of time.
3. **Performance :**
  - a. JavaScript can be written such that the HTML files are fairly compact and quite small.
  - b. It minimizes storage requirements on the web server and download time for the client.
4. **Easy debugging and testing :** Being an interpreted language, scripts in JavaScript are tested line by line and the errors are also listed as they are encountered.

## Web Technology

- 5. Platform independence/architecture neutral :**
- JavaScript is completely independent of the hardware on which it works.
  - It is a language that is understood by any JavaScript enabled browser.

**Weakness/Disadvantages of JavaScript :**

- There are issues of incompatibility of several scripting that result in website overloading.
- Different layout engines may render JavaScript differently resulting in inconsistency in terms of functionality and interface.
- JavaScript is also a common tool for the web hackers, they use script injection to hack a site.
- JavaScript is light, somehow too much of JavaScript can slow down the page loading of a website.

**Que 3.3.** Write short notes on JavaScript DOM.

**Answer**

- A document object represents the HTML document that is displayed in the window.
- In DOM (Document Object Model), the document object has various properties that refer to other objects which allow access to and modification of document content.
- The hierarchical structure of object is applied for the organization of objects in a web document which include following object :
  - Window object :** It is the top most element of the object hierarchy.
  - Document object :** Each HTML document that gets loaded into a window becomes a document object which contains the contents of the page.
  - Form object :** Everything enclosed in the `<form>...</form>` tag sets the form object. The form object contains all the elements defined for that object such as text fields, buttons, radio buttons and checkboxes.
- When a web page is loaded, the browser creates a Document Object Model of the page.
- DOM is an object oriented representation of an HTML document and acts as an interface between JavaScript and the document itself and allows the creation of dynamic web pages.
- DOM supports navigation in any direction (i.e., parent and previous sibling) and allows for arbitrary modifications.

**Que 3.4.** What is the difference between Java and JavaScript ? Write a JavaScript function for e-mail address validation, that is to check if the content has the general syntax of an e-mail or not.

**AKTU 2015-16, Marks 10**

**Answer**

**Difference between Java and JavaScript :** Refer Q. 3.2, Page 3-4Z, Unit-3.

**JavaScript function for e-mail address validation :**

```

<html>
<head>
<title> The Student Registration Form </title>
<script type="text/javascript">
function validate()
{
    var i;
    var str=document.my_form.Email_txt.value;
    var index_at=str.indexOf("@");
    var len=str.length;
    var index_dot=str.indexOf(".");
    var emailID=document.my_form.Email_txt;
    if((emailID.value==null) | |(emailID.value == " "));
    // user has not given Email ID
    {
        alert("Please Enter your Email ID");
        emailID.focus();
        return false;
    }
    if(str.indexOf("@") == -1); // there is no @ symbol in email address
    {
        alert("Invalid Email ID");
        return false;
    }
    if(str.indexOf(".") == -1 | | str.indexOf(".") == 0 | |
    str.indexOf(".") == index_at)
    {
        alert("Please Enter valid Email ID");
        emailID.focus();
        return false;
    }
}

```

```

// there is no dot in the email ID
{
    alert("Invalid Email ID");
    return false;
}

if(str.indexOf("@", (index_at+1)) != -1); // no text after @ symbol
{
    alert("Invalid Email ID");
    return false;
}

if(str.indexOf("") != -1); // blank space in the email ID
{
    alert("Invalid Email ID");
    return false;
}

return true;
}

```

</script>

</head>

</html>

**Que 3.5.** How do you perform client-side validation using JavaScript? Illustrate with suitable example.

**Answer**

Following are the steps used to perform client-side validation using JavaScript :

1. First the user will enter the value in the form field.
2. Then, browser will ensure that the value provided by user is correct and valid so that successful validation can be done.
3. JavaScript used in the web page uniquely defines all the special functionalities in the client browser.
4. By default, if there is a validation error then an error or pop-up message is shown by the browser.
5. If there is no error then the validation on client-side will be successfully performed.

AKTU 2014-15, Marks 06

**For example :**

If a form field (fname) is empty, this function alerts a message, and returns false, to prevent the form from being submitted:

```

function validateForm() {
    var x = document.forms["myForm"]["fname"].value;
    if (x == "") {
        alert("Name must be filled out");
        return false;
    }
}

```

The JavaScript function is called when the form is submitted :

```

<form name="myForm" action="/action_page.php" onsubmit="return validateForm()" method="post">
    Name: <input type="text" name="fname">
    <input type="submit" value="Submit">
</form>

```

**Que 3.6.** What are scripting languages and why JavaScript is used ? Write a JavaScript function for validating form data like mandatory fields and email field.

AKTU 2017-18, Marks 05

**Answer**

Scripting language :

1. A scripting language is a programming language designed for integrating and communicating with other programming languages.
2. Some of the most widely used scripting languages are JavaScript, VBScript, PHP, Perl, Python, Ruby, ASP.

JavaScript is used because :

- a. It is executed on client side.
- b. It saves bandwidth on web server.
- c. It is written into an HTML page.

JavaScript function for validating form data :

```

function formValidation()
{
    var uid = document.registration.userid;
    var passid = document.registration.passid;
    var uname = document.registration.username;
    var uaddr = document.registration.address;
    var ucountry = document.registration.country;
}

```

```

var uzip = document.registration.zip;
var uemail = document.registration.email;
var umsex = document.registration.msex;
var ufsex = document.registration.fsex;
if(userid_validation(uid,5,12))
{
if(passid_validation(passid,7,12))
{
ifallLetter(uname)
{
ifalphanumeric(uadd)
{
ifcountryselect(ucountry)
{
ifallnumeric(uzip)
{
ifValidateEmail(uemail)
{
ifvalidsex(umsex,ufsex)
{
}
}
}
}
}
}
return false;
}
function userid_validation(uid,mx,my)
{
var uid_len = uid.value.length;
if(uid_len == 0 || uid_len >= my || uid_len < mx)
alert("User Id should not be empty / length be between "+mx+" to "+my);
uid.focus();
return false;
}
return true;
}
function passid_validation(passid,mx,my)
{
var passid_len = passid.value.length;
if(passid_len == 0 || passid_len >= my || passid_len < mx)
{
alert("Password should not be empty / length be between "+mx+" to "+my);
}
}

```

```

passid.focus();
return false;
}
return true;
}
function allLetter(uname)
{
var letters = /^[A-Za-z]+$/;
if(uname.value.match(letters))
{
return true;
}
else
{
alert('Username must have alphabet characters only');
uname.focus();
return false;
}
}
function alphanumeric(uadd)
{
var letters = /^[0-9a-zA-Z]+$/;
if(uadd.value.match(letters))
{
return true;
}
else
{
alert('User address must have alphanumeric characters only');
uadd.focus();
return false;
}
}
function countryselect(ucountry)
{
if(ucountry.value == "Default")
{
alert('Select your country from the list');
ucountry.focus();
return false;
}
else
{
return true;
}
}
function allnumeric(uzip)
{
}

```

```

{
var numbers = /^[0-9]+$/;
if(uzip.value.match(numbers))
{
return true;
}
else
{
alert('ZIP code must have numeric characters only');
uzip.focus();
return false;
}
}

function ValidateEmail(uemail)
{
var mailformat = /^[^\w+(\[.\-]\?\w+)*@\w+(\[.\-]\?\w+)*(\.\w{2,})$/;
if(uemail.value.match(mailformat))
{
return true;
}
else
{
alert("You have entered an invalid email address!");
uemail.focus();
return false;
}
}

function validsex(umsex,ufsex)
{
x=0;
if(umsex.checked)
{
x++;
}
if(ufsex.checked)
{
x++;
}
if(x==0)
{
alert('Select Male/Female');
umsex.focus();
return false;
}
else
{
alert('Form Successfully Submitted');
}
}

```

```

window.location.reload()
return true; }

```

**Que 3.7.** Explain conditional statements used in JavaScript with example.

**Answer**

There are following conditional statement used in JavaScript :

1. **If statement :** If statement is used if we want to execute some code only if a specified condition is true.

**Syntax :**

```

if(condition)
{
    code to be executed if condition is true
}

```

**Example :**

```

<script type = "text/javascript">
var d = new Date();
var time = d.getHours();
if(time<10)
{
document.write ("<b>Good morning to all</b>");
}
</script>

```

2. **If...else statement :** If...else statement is used when we do not confirm about the condition that is true or not.

**Syntax :**

```

if(condition)
{
    code to be executed if condition is true
}
else
{
    code to be executed if condition is not true
}

```

**Example :**

```

<script type = "text/javascript">
//If the time is less than 10, print "Good Day" Otherwise "Good Night"
var d = new Date ();

```

```

var time = d.getHours();
if(time < 10)
{
    document.write("Good Day");
}
else
{
    document.write("Good Night");
}
</script>
3. If...else if...else statement : We can use the if...else if...else statement
if we want to select one from many sets of lines.
Syntax :
if(condition1)
{
    code to be executed if condition1 is true
}
else if(condition2)
{
    code to be executed if condition2 is true
}
else
{
    code to be executed if condition1 and condition2 are not true
}
Example :
<script type = "text/javascript">
var d = new Date();
var time = d.getHours();
if(time<10)
{
    document.write("<b>Good morning</b>");
}
else if(time>10 && time<16)
{
    document.write("<b>Good afternoon</b>");
}
else
{
}

```



```

document.write("<b>Good night</b>");
}
</script>

```

4. **Switch statement :** If we want to select one of many blocks of code then we use switch statement.

**Syntax :**

```

switch(n)
{
    case 1:
        execute code block 1
        break;
    case 2:
        execute code block 2
        break;
    default:
        code to be executed if n is different from case 1 and 2
}

```

**Example :**

```

<script type = "text/javascript">
var d=new Date();
theDay=d.getDay();
switch(theDay)
{
    case 1:
        document.write("Finally Monday");
        break;
    case 2:
        document.write("Super Tuesday");
        break;
    case 0:
        document.write("Sleepy Sunday");
        break;
    default:
        document.write("I'm looking forward to this weekend!");
}
</script>

```



**Que 3.8.** Describe data types, functions and objects used in JavaScript with an example.

**Answer**

**Data types in JavaScript :** JavaScript provides different data types to hold different types of values.

There are two types of data types in JavaScript :

1. **Primitive data type :** There are five types of primitive data types in JavaScript. They are as follows :

Data type	Description
String	It represents sequence of characters.
Number	It represents numeric.
Boolean	It represents boolean value either false or true
Undefined	It represents undefined value
Null	It represents null i.e., no value at all

2. **Non-primitive (reference) data type :** The non-primitive data types are as follows :

Data Type	Description
Object	It represents instance through which we can access members
Array	It represents group of similar values
RegExp	It represents regular expression

**Function in JavaScript :**

1. Functions can be defined both in the <head> and in the <body> section of a document.
2. However, to assure that the function is read/loaded by the browser before it is called, it is needed to be defined in the <head> section.
3. **Syntax :**

```
function function_name (var1, var2, ..., varX)
{
    some code
}
```

The curly braces {} and the } defines the start and end of the function.

4. A function with no parameters must include the parentheses () after the function name :

```
function function_name ()
{
    some code
}
```

**Objects in JavaScript :**

1. **Built-in objects :** These objects are used quite extensively for data processing in JavaScript. Following are some built-in object :

- a. **String object :**
  - i. The string object enables programs to work with and manipulate string.
  - ii. It provide methods such as : big(), blink(), bold(), italics(), charAt(), touppercase(), tolowercase() and substring().
- b. **Math object :**
  - i. The math object provides some commonly used methods such as : sqrt(num), abs(num), sin(num), cos(num), tan(num), exp(num), min(a, b), max(a, b), log(num), pow(a, b), floor(num), ceil(num) etc.
- c. **Date object :**
  - i. The date object enables JavaScript programmers to create an object that contains information about a particular date and provides a set of methods to work with that information.
  - ii. **Syntax :**

```
var mydate = new Date(<parameters>);
```

- iii. If the parameter left empty, it indicates current date and time.
- iv. The date object provides some methods which are : getDate(), setDate(), getHours(), setHours(), getTime(), setTime(), getDay(), setDay(), getMinutes(), setMinutes(), getSecond(), setSecond().

- d. **Array object :**

- i. The array object stores multiple values in a single variable.
- ii. **Syntax :**

```
var fruits = new Array("apple", "orange", "mango");
```

2. **User-defined objects :**

- a. A user-defined object is also associated with properties and methods, which belong to it.
- b. The user-defined object would also require methods that will allow the storage of name, age and salary of the employee object.

```
function Employee(name, age, salary)
```

## 3-17 Z (CS-5/IT-6)

## Web Technology

- ```

        this.name = name;
        this.age = age;
        this.salary = salary;
    }

c. In the given example, object Employee has three properties : name, age and salary. Here, this keyword refers to the current object in focus. In given example, this.name will refer to the name of the current object.
  
```
- Que 3.9.** How do you associate functions with objects using JavaScript ?

**Answer**

1. In JavaScript we can associate functions with objects by creating objects and assigning properties.
2. Every object in JavaScript has different property.
3. Property of an object is defined in a variable which is attached to an object.
4. Object properties are mostly the same as simple JavaScript variables.
5. JavaScript has a number of predefined objects. So, we can create our own objects and embed the function directly in the object.

**Example :**

```

var myCar = new Object();
myCar.make = "Nissan";
myCar.model = "N-54";
myCar.year = 1972;
<script type="text/javascript">
function user(name, email) {
    this.name = name;
    this.email = email;
    //Custom method for object
    this.toString = function userToString() {
        return("Name: "+this.name+" Email: "+this.email);
    }
}
var obj = new user("Aditya", "aditya123@example.com");
document.write(obj.toString());
</script>
  
```

## 3-18 Z (CS-5/IT-6)

- Que 3.10.** What is dynamic HTML ? How DHTML is used with JavaScript ?

**AKTU 2014-15, Marks 06****Answer**

Dynamic HTML : Refer Q. 2.27, Page 2-35Z, Unit-2.

DHTML used in JavaScript as follows :

1. DHTML is used with JavaScript through following syntax :
- ```

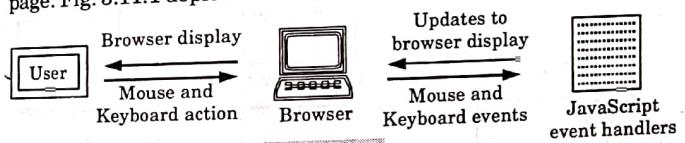
<script type="text/Javascript">
...
</script>
  
```
2. DHTML makes use of CSS, events, methods and so on to make the page dynamic.

- Que 3.11.** How do we handle event in JavaScript ? What is DHTML ?

**AKTU 2013-14, Marks 06****Answer**

## Event handling in JavaScript :

1. JavaScript allows us to change the default action associated with events. For example, event processing where a user clicks a hyperlink on a page. Fig. 3.11.1 depicts how events occur and are handled in JavaScript.

**Fig. 3.11.1.**

2. Clicking on the hyperlink generates an event and the default associated action, then browser loads and displays the page associated with that URL.
3. However, in JavaScript, we can change this default action and write our own event handler that will be associated with this hyperlink.
4. The following are a few other tasks that we can do with events using JavaScript event handlers :
  - a. Validate the data entered by a user in a form.
  - b. Shift the focus of controls from one field to another in a form.
  - c. Load and display animation when a user clicks a button.
  - d. Communicate with Java applets and browser plug-ins.
  - e. Display a dialog box when a user moves the mouse over a link.

DHTML : Refer Q. 2.27, Page 2-35Z, Unit-2.

**Que 3.12.** Write a short note on AJAX.

AKTU 2013-14, 2016-17; Marks 03

**Answer**

1. AJAX (Asynchronous JavaScript and XML) is a set of web development techniques for creating better, faster and more interactive web applications with the help of XML, HTML, CSS and JavaScript.
2. Traditional web applications tend to follow the pattern shown in Fig. 3.12.1.
3. First a page is loaded. Next, the user performs some action such as filling out a form or clicking a link.
4. The user activity is then submitted to a server-side program for processing while the user waits until final result is sent which reloads the entire page.
5. AJAX style applications use a significantly different model. Here user actions signal the server to fetch just the data needed to update the page in response to the submitted actions.
6. This process generally happens asynchronously, thus it allows the user to perform other actions within the browser while data is returned.

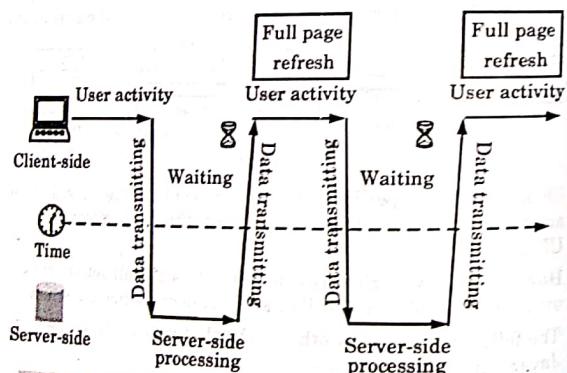


Fig. 3.12.1. Traditional web application communication flow.

7. Asynchronous requests allow more than one thing to happen at the same time.
8. Only the relevant portion of the page is changed when we use AJAX, as shown in Fig. 3.12.2.

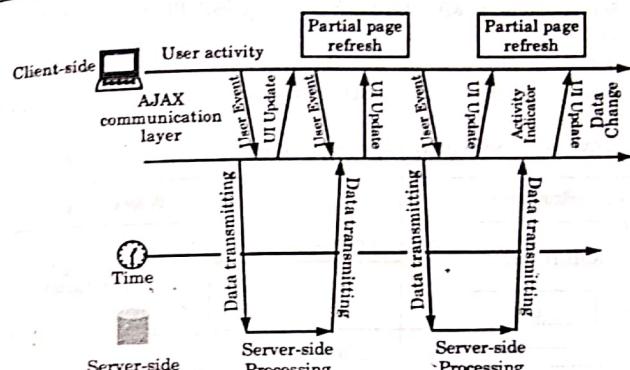


Fig. 3.12.2. AJAX style communication flow.

**Que 3.13.** What is AJAX? Explain its advantage and its working.

Explain with example.

AKTU 2017-18, Marks 10

**Answer**

AJAX : Refer Q. 3.12, Page 3-19Z, Unit-3.

**Advantages of AJAX:**

1. **Better interactivity :** AJAX allows easier and quicker interaction between user and website as pages are not reloaded for content to be displayed.
2. **Easier navigation :** AJAX applications on websites can be built to allow easier navigation to users in comparison of using the traditional back and forward button on a browser.
3. **Compact :** With AJAX, several multi-purpose applications and features can be handled using a single web page, avoiding the need for clutter with several web pages.
4. **Backed by reputed brands :** Several complex web applications are handled using AJAX like Google maps.

**Working of AJAX :**

1. XMLHttpRequest object plays an important role as AJAX communicates with the server using XMLHttpRequest object.
2. User sends a request from the UI and a JavaScript call goes to XMLHttpRequest object.
3. HTTP request is sent to the server by XMLHttpRequest object.

- Web Technology**
4. Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
  5. Data is retrieved.
  6. Server sends XML data or JSON data to the XMLHttpRequest callback function.
  7. HTML and CSS data is displayed on the browser.

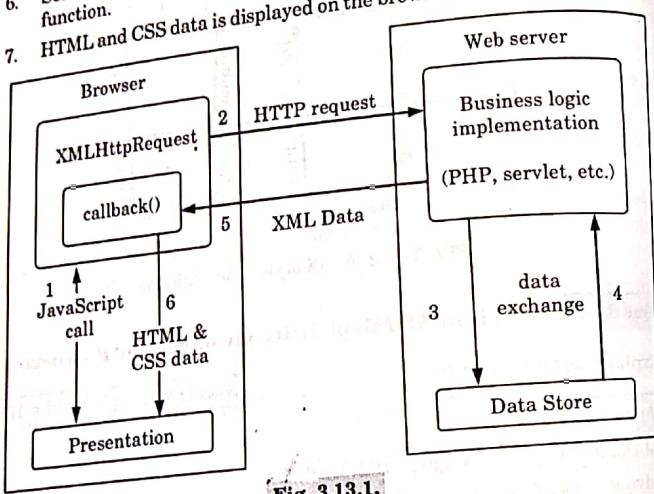


Fig. 3.13.1.

**Example :**

```

<!DOCTYPE html>
<html>
<body>
<div id="demo">
<h2>The XMLHttpRequest Object</h2>
<button type="button" onclick="loadDoc()">Change Content</button>
</div>
<script>
function loadDoc() {
var xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if(this.readyState == 4 && this.status == 200) {
document.getElementById("demo").innerHTML = this.responseText;
}
};
xhttp.open("GET", "ajax_info.txt", true);
xhttp.send();
}

```

```

</script>
</body>
</html>

```

**Que 3.14.** What is AJAX ? Explain the application of AJAX with the help of suitable examples.

**AKTU 2015-16, Marks 10**

**Answer**

AJAX : Refer Q. 3.12, Page 3-19Z, Unit-3.

**Applications of AJAX are :**

1. AJAX is used to change the text without reloading the web page.
2. AJAX is a technique used for creating fast and dynamic web pages.
3. AJAX contains div section which is used to display information returned from a server.

**For example :**

```

<html>
<head>
<script type="text/javascript">
function load XML DOC()
{
//AJAX script goes here
}
</script>
</head>
<body>
<div id = "myDiv"> Let AJAX change this text</div>
<button type = "button" onclick = "Load XML DOC()"> change content</button>
</body>
</html>

```

**PART-2**

**Networking :** Internet Addressing, InetAddress, Factory Method, Instance Methods, TCP/IP Client Sockets, URL, URL Connection, TCP/IP Server Socket, Datagram.

**CONCEPT OUTLINE : PART-2**

- Internet address is a number that uniquely identifies each computer on network.
- InetAddress is a class used to encapsulate both numerical IP address and the domain name.
- Factory methods help us to create InetAddress object.
- Datagrams are bundles of information passed between machines.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 3.15.** Explain IP addressing.

**Answer**

- The IP address is a network layer address that uniquely identifies each computer on network.
- Each TCP/IP host is identified by a logical IP address.
- A unique IP address is required for each host and network component that communicates using TCP/IP.
- The IP address identifies a system's location on the network. An IP address must be globally unique and have a uniform format.
- Each IP address includes a network ID and a host ID.
  - The network ID (also known as a network address) identifies the systems that are located on the same physical network ID. The network ID must be unique to the internetwork.
  - The host ID (also known as a host address) identifies a workstation, server, router, or other TCP/IP host within a network. The address for each host must be unique to the network ID.
- The use of the term network ID refers to any IP network ID, whether it is class-based, a subnet, or a supernet.

An internet address is made of four bytes (32 bits) that define a host's connection to a network.

Class type	Netid	Hostid
------------	-------	--------

**Fig. 3.15.1.**

- An IP address is 32 bits long. It is a common practice to segment the 32 bits of the IP address into four 8-bit fields called octets.
- Each octet is converted to a decimal number (the base 10 numbering system) in the range 0-255 and separated by a period (a dot). This formal is called dotted decimal notation.

**Que 3.16.** Give the classification of different IP address.

**Answer**

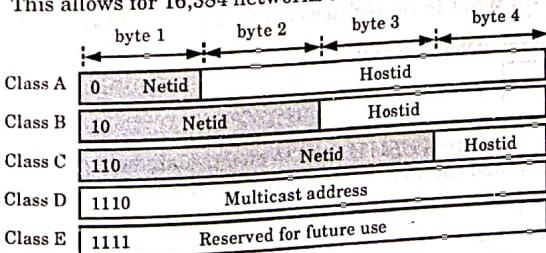
IP address is classified as :

**1. Class A :**

- Class A addresses are assigned to networks with a very large number of hosts.
- The high-order bit in a class A address is always set to zero.
- The next seven bits (completing the first octet) complete network ID. The remaining 24 bits (the last three octets) represent the host ID.
- This allows for 126 networks and 16,777,214 i.e.,  $2^{24}$  hosts per network.

**2. Class B :**

- Class B addresses are assigned to medium-sized to large-sized networks.
- The two high-order bit in a class B address are always set to binary 10.
- The next 14 bits (completing the first two octets) complete the network ID. The remaining 16 bits (last two octets) represent the host ID.
- This allows for 16,384 networks and 65,534 hosts per network.



**Fig. 3.16.1.**

**3. Class C :**

- Class C addresses are used for small networks.
- The three high-order bits in a class C address are always set to binary 110.

- iii. The next 21 bits (completing the first three octets) complete the network ID. The remaining 8 bits (last octet) represent the host ID.
- iv. This allows for 2,097, 152 networks and 254 hosts per network.

	From	To
Class A	0.0.0.0	127.255.255.255
	Netid Hostid	Netid Hostid
Class B	128.0.0.0	191.255.255.255
	Netid Hostid	Netid Hostid
Class C	192.0.0.0	233.255.255.255
	Netid Hostid	Netid Hostid
Class D	224.0.0.0	239.255.255.255
	Group address	Netid Hostid
Class E	240.0.0.0	255.255.255.255
	Undefined	Undefined

Fig. 3.16.2.

**4. Class D :**

- i. Class D addresses are reserved for IP multicast addresses.
- ii. The four high-order bits in a class D address are always set to binary 1110.
- iii. The remaining bits are for the address that interested hosts will recognize.
- iv. Microsoft supports class D addresses for applications to multicast data to multicast-capable hosts on an internetwork.
- 5. Class E : Class E addresses are experimental addresses reserved for future use. The high-order bits in a class E address are set to 1111.

**Que 3.17.** What is InetAddress class ? Explain factory method of InetAddress with example.

**Answer****InetAddress class :**

1. The InetAddress class is used to encapsulate both the numerical IP address and the domain name for that address.
2. The InetAddress class hides the number inside.
3. InetAddress can handle both IPv4 and IPv6 addresses.
4. The InetAddress class has no visible constructors.

**Factory methods :**

1. Factory methods are used to create an InetAddress object.
2. Factory methods are convention whereby static methods in a class return an instance of that class. This is done instead of overloading a

constructor with various parameter lists when having unique method names makes the results much clear.

3. Three commonly used InetAddress factory methods are as follows :

- a. **The getLocalHost()** : This method simply returns the InetAddress object that represents the local host.
- b. **The getByName()** : This method returns an InetAddress for a host name passed to it. If this method is unable to resolve the host name then, they throw an UnknownHostException.
- c. **The getAllByName()** : This method returns an array of InetAddresses that represent all of the addresses that a particular name resolves to. It will also throw an UnknownHostException if it cannot resolve the name to at least one address.

**For example :** The following example prints the addresses and names of the local machine and two well-known Internet websites :

// Demonstrate InetAddress.

```
import java.net.*;
class InetAddressTest
{
    public static void main(String args[]) throws UnknownHostException
    {
        InetAddress Address = InetAddress.getLocalHost();
        System.out.println(Address);
        Address = InetAddress.getByName("osborne.com");
        System.out.println(Address);
        InetAddress SW[] = InetAddress.getAllByName("www.nba.com"); for
        (int i=0; i<SW.length; i++)
        System.out.println(SW[i]);
    }
}
```

**Output :**

```
default/206.148.209.138
osborne.com/198.45.24.162
www.nba.com/64.241.238.153
www.nba.com/64.241.233.142
```

**Que 3.18.** Write short note on instance method.

**Answer**

1. Instance method is a method defined in a class and only accessible through the object of the class.

## Web Technology

3-27 Z (CS-5/IT-6)

2. The InetAddress class has several instance methods, which can be used on the objects.
3. Following are the object returned by the methods which are as follows:
- Boolean equals(Object other) :** It returns true if this object has the same Internet address as other. Otherwise, it returns false.
  - Byte[ ] getAddress() :** It returns a byte array that represents the object's Internet address in network byte order.
  - String getHostAddress() :** It returns a string that represents the host address associated with the InetAddress object.
  - String getHostName() :** It returns a string that represents the host name associated with the InetAddress object.
  - String toString() :** It returns a string that lists the host name and the IP address.

**Que 3.19.** Explain TCP/IP client socket. Also, write the constructor and methods used to create a client socket.

**Answer**

- TCP/IP client sockets are used to implement reliable, bi-directional, persistent, point-to-point, stream-based connections between hosts on the Internet.
- A socket can be used to connect Java I/O system to other programs that may reside either on the local machine or on any other machine on the Internet.
- The creation of a Socket object implicitly establishes a connection between the client and server.
- Following are the two constructors used to create client socket:
  - Socket(String hostName, int port) :** Creates a socket connecting the local host to the named host, port and can throw an UnknownHostException or an IOException.
  - Socket(InetAddress ipAddress, int port) :** Creates a socket using a pre-existing InetAddress object, a port and can throw an IOException.
- Following methods are used by TCP/IP client socket:
  - InetAddress getInetAddress() :** Returns the InetAddress associated with the Socket object.
  - Int getPort() :** Returns the remote port to which the Socket object is connected.
  - Int getLocalPort() :** Returns the local port to which the Socket object is connected.

Scripting and Networking

3-28 Z (CS-5/IT-6)

**Que 3.20.** Write short note on URL and URLConnection class in Java.

**Answer**

- URL is an acronym for Uniform Resource Locator.
- It points to a resource on the World Wide Web (WWW).
- A URL contains many information like protocol name, server name, port number and file name.
- The URL is represented by an URL class.
- Consider the following URL:  
<http://www.quantumpage.com/aktu-paper.html>
  - Protocol :** In this case, http is the protocol.
  - Server name or IP address :** In this case, www.quantumpage.com is the server name.
  - Port number :** It is an optional attribute. If we write <http://www.quantumpage.com:80/aktu-papers.html>, 80 is the port number. If port number is not mentioned in the URL, it returns -1.
  - File name or directory name :** In this case, aktu-papers.html is the file name.

Following are the method provided by java.net.URL class :

S.No.	Method	Description
1.	public String getProtocol()	it returns the protocol of the URL.
2.	public String getHost()	it returns the host name of the URL.
3.	public String getPort()	it returns the Port Number of the URL.
4.	public String getFile()	it returns the file name of the URL.
5.	public URLConnection openConnection()	it returns the instance of URLConnection i.e., associated with this URL.

**URLConnection class :**

- The Java URLConnection class represents a communication link between the URL and the application.
- This class can be used to read and write data to the specified resource referred by the URL.
- The openConnection() method of URL class returns the object of URLConnection class.

## Web Technology

4. Syntax to get the object of URLConnection :  
public URLConnection openConnection() throws IOException()
5. The URLConnection class uses `getInputStream()` method to display all the data of a web page.
6. The `getInputStream()` method returns all the data of the specified URL in the stream that can be read and displayed.

**Que 3.21.** Explain TCP/IP server socket.

### Answer

1. The TCP/IP ServerSocket is used to create servers that listen for either local or remote client programs to connect them on published ports.
2. TCP/IP ServerSockets are quite different from normal sockets.
3. When we create a TCP/IP Server Socket, it will register itself with the systems that have client connections.
4. The constructors for TCP/IP ServerSocket reflect the port number that we wish to accept connections on and, how long we want the port to be in the queue.
5. The queue length tells the system how many client connections it can leave pending before it should simply refuse connections.
6. It has constructors that create new TCP/IP ServerSocket objects, methods that listen for connections on a specified port, methods that configure the various TCP/IP server socket options, and the usual miscellaneous methods such as `toString()`.

**Que 3.22.** What is datagram? Give its characteristics. Also, explain datagram socket.

### Answer

1. Datagram is a unit of transfer associated with networking.
  2. Datagram is typically structured in header and payload section.
  3. It provides a connectionless communication service across a packet-switched network.
- Characteristics of datagram :**
1. It is transmitted from source to destination without guarantee of delivery.
  2. It provides a connectionless communication service.
- Datagram socket :**
1. It is a communication link used to send datagrams between applications.
  2. Datagram socket is a type of network socket which provide connectionless point for sending and receiving packets.

3-29 Z (CS-5/IT-6)

## Scripting and Networking

3-30 Z (CS-5/IT-6)

3. Every packet sent from a datagram socket is individually routed and delivered.
4. Java DatagramSocket and DatagramPacket classes are used for connectionless socket programming.
5. A DatagramPacket is a message that can be sent or received through DatagramSocket.
6. Commonly used constructors of DatagramSocket class are as follows :
  - a. `DatagramSocket()` throws `SocketException` : It creates a datagram socket and binds it with the available port number on the localhost machine.
  - b. `DatagramSocket(int port)` throws `SocketException` : It creates a datagram socket and binds it with the port number.
  - c. `DatagramSocket(int port, InetAddress address)` throws `SocketException` : It creates a datagram socket and binds it with the specified port number and host address.
7. Commonly used constructors of DatagramPacket class are as follows :
  - a. `DatagramPacket(byte[] barr, int length)` : It creates a datagram packet. This constructor is used to receive the packets.
  - b. `DatagramPacket(byte[] barr, int length, InetAddress address, int port)` : It creates a datagram packet. This constructor is used to send the packets.

## VERY IMPORTANT QUESTIONS

Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.

**Q. 1. What is JavaScript ? How it works ?**

**Ans:** Refer Q. 3.1.

**Q. 2. Write short notes on JavaScript DOM.**

**Ans:** Refer Q. 3.3.

**Q. 3. How do you perform client-side validation using JavaScript ?**

**Ans:** Refer Q. 3.5.

**Q. 4. Write a short note on :**  
a. Object in JavaScript

**PART-1**

**Enterprise Java Bean : Preparing a Class to be a Java Beans.**  
**Creating a Java Beans, Java Beans Properties, Types of Beans,**  
**Stateful Session Bean, Stateless Session Bean, Entity Bean.**

**CONCEPT OUTLINE : PART-1**

- EJB is a server-side component.
- Java Bean is a software component that is used in variety of different environments.

**Questions-Answers****Long Answer Type and Medium Answer Type Questions**

**Que 4.1.** What is EJB ? Write the advantages and disadvantages of EJB.

**Answer**

- An Enterprise Java Bean is a server-side component which encapsulates business logic.
- EJB (Enterprise Java Bean) is used to develop scalable, robust and secured enterprise applications in Java.
- Middleware services such as security, transaction management etc. are provided by EJB container to all EJB applications.
- To run EJB application, we need an application server (EJB Container) such as Jboss, Glassfish, Weblogic, Websphere etc.
- It performs :
  - Life cycle management
  - Security
  - Transaction management
  - Object pooling
- EJB application is deployed on the server, so it is also called server-side component.

**Advantages of EJB :**

- EJB provides developers an architectural independence.
- EJB establishes roles for application development.

**Web Technology**

- EJB provides distributed transaction support.
- It provides portable and scalable solutions of the problem.
- It provides of vendor specific enhancements.

**Disadvantages of EJB :**

- It requires application server.
- It requires only Java client. For other language client, you need to go for web service.
- It is complex to understand and develop EJB applications.

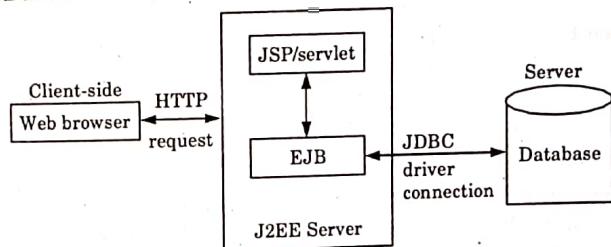
**Que 4.2.** Explain EJB architecture. What are its various types ? Describe the steps used to create Java Bean and to build application using BDK.

AKTU 2015-16, Marks 10

**OR**

Write short note on EJB architecture.

AKTU 2016-17, Marks 05

**Answer****EJB architecture :****Fig. 4.2.1. Architecture.**

The EJB architecture is an extension of web architecture.

**Working of EJB architecture :**

- The client is working on a web browser.
- There is a database server that hosts a database, like MySQL / Oracle.
- The J2EE server machine is running on an application server.
- The client interface is provided with JSP / Servlet.
- The application server manages the relationships between the client and database.

**Types of EJB :**

- Entity bean :** Entity beans represent persistent data storage. Entity beans are used for modeling the business concept.

- Session bean :** Session beans are used for managing processes or tasks. Hence, session beans are used for managing activities.
- Message driven bean :** Message driven bean is similar to the session bean but it gets activated only when asynchronous message arrives. When a message arrives then the EJB container calls the message driven bean on message method to process the message.

**Steps used to create Java Bean :**

Step 1 : Put source code into a file named "SimpleBean.java" :

```
import java.awt.*;
import java.io.Serializable;
public class SimpleBean extends Canvas
implements Serializable {
//Constructor sets inherited properties
public SimpleBean () {
setSize(60, 40);
setBackground(Color.red);
}
}
```

Step 2 : Compile the file :

```
javac SimpleBean.java
```

Step 3 : Create a manifest file, named "manifest.tmp" :

```
Name: SimpleBean.class
Java-Bean: True
```

Step 4 : Create the JAR file, named "SimpleBean.jar" :

```
jar cfm SimpleBean.jar manifest.tmp SimpleBean.class
```

Then, verify that the content is correct by the command "jar tf

Step 5 :

1. Start and run the Bean Box.
2. Load JAR file into Bean Box by selecting "Loadjar..." under the File menu.

Step 6 :

1. After the file selection dialog box is closed, Then "SimpleBean" appear at the bottom of the toolbox window.

Select SimpleBean.jar.

Cursor will change to a plus. In the middle BeanBox window, we can now click to drop in what will appear to be a coloured rectangle. Step 7 : Try changing the red box colour with the Properties windows.

Step 8 : Choose "Events" under the "Edit" menu in the middle window to see what events SimpleBean can send. These events are inherited from jav.awt.Canvas.

**Steps to build application using BDK :**

Step 1 : Create a directory for the new bean.

Step 2 : Create the Java source file(s).

Step 3 : Compile the source file(s).

Step 4 : Create a manifest file.

Step 5 : Generate a JAR file.

Step 6 : Start the BDK.

Step 7 : Test the newly created Java Bean.

**Que 4.3. What is a Java Bean ? Why they are used ? Explain the properties of Java Beans.**

**Answer**

1. A Java Bean is a software component that has been designed to be reused in a variety of different environments.
2. It may perform simple functions such as checking the spelling of a document, or a complex function such as forecasting the performance of a stock portfolio.
3. A Bean may be visible to an end user.
4. A Bean may be designed to work autonomously on a user's workstation or to work in cooperation with a set of other distributed components.

**Java Beans are used because :**

1. It encapsulates many objects into a single object.
2. It allows us to use properties of getter and setter methods.
3. It has Java object which has nullary constructor.
4. It can be manipulated visually in a builder tools.

**Three types of properties are :****1. Simple properties :**

- a. A simple property has a single value.
  - b. It can be identified by the following design patterns, where *N* is the name of the property and *T* is its type.
- ```
public T getN();
public void setN(T parameter)
```
- c. If the property has both read and write permission then both the get and set methods can access the values. Otherwise, only one method can access the values.

- d. If the property has only read permission then only get method can access the values, similarly if the property has only write permission then only set method can access the values.

**Boolean properties :**

- A boolean property has a value of true or false.
- It can be identified by the following design patterns, where  $N$  is the name of the property.  

```
public boolean isN();
public void setN(boolean parameter);
public Boolean getN();
```
- For getting the values isN and getN methods are used and for setting the Boolean values setN method is used.

**Indexed properties :**

- An indexed property consists of multiple values.
- It can be identified by the following design patterns, where  $N$  is the name of the property and  $T$  is its type.  

```
public T getN(int index);
public void setN(int index, T value);
public T[] getN();
public void setN(T values[]);
```

**Que 4.4. How to prepare a class to be a Java Beans ?**

**Answer**  
 Java Beans are classes that encapsulate many objects into a single object. It is a Java class that should have following conventions:  
 1. It must implement serializable interface.  
 2. It should have a public constructor without argument.  
 3. All properties in Java Bean must be private with public getters and setter methods.

**Example of Java Bean class :**  

```
//Employee.java
package mypack;
public class Employee implements java.io.Serializable{
    private int id;
    private String name;
    public Employee(){}
    public void setId(int id){this.id=id;}
    public int getId(){return id;}}
```

```
public void setName(String name){this.name=name;}
public String getName(){return name;}
```

To access the Java Bean class, we should use getter and setter methods :

```
package mypack;
public class Test{
    public static void main(String args[]){
        Employee e=new Employee(); //object is created
        e.setName("Arjun"); //setting value to the object
        System.out.println(e.getName()); // getting value to the object
    }}
```

**Que 4.5. Explain session beans with its types.****Answer****Session bean :**

- Session bean encapsulates business logic only, it can be invoked by local, remote and web service client.
- It can be used for managing activities like database access, calculation etc.
- The life cycle of session bean is maintained by the application server (EJB container).
- Session bean is created by a customer and its duration is only for the signal client server session.
- This bean can be transactional, because it is not recoverable and due to this a system crash can occur.
- Its objective is to manage its own unrelenting data.

**Types of session bean :**

- Stateless session bean :**
  - Stateless session bean is a business object that represents business logic only. It does not have state (data).
  - Conversational state between multiple method calls is not maintained by the container in case of stateless session bean.
  - The stateless bean objects are pooled by the EJB container to service the request on demand.
  - It can be accessed by one client at a time. In case of concurrent access, EJB container routes each request to different instance.
  - The stateless session beans is distributed object which has no connection with informal state; only allow parallel access to beans.

f. Annotations used in stateless session bean are :

- i. @Stateless
- ii. @PostConstruct
- iii. @PreDestroy

**Stateful session bean :**

- a. Stateful session bean is a business object that represents business logic like stateless session bean. But, it maintains state (data).
- b. Conversational state between multiple method calls is maintained by the container in stateful session bean.
- c. There are five important annotations used in stateful session bean:
  - i. @Stateful
  - ii. @PostConstruct
  - iii. @PreDestroy
  - iv. @PrePassivate
  - v. @PostActivate

**Que 4.6.** Describe entity beans with its types.

**Answer**

- Entity beans are objects that represent a persistence storage mechanism.
- Each entity beans has underlying table in a relational database and each row in the table represents the instance of the bean.

There are two types of entity beans :

**1. Container Managed Persistence (CMP) :**

- a. The term Container Managed Persistence means that the EJB container handles all database access required by the entity bean.
- b. The bean code contains no database access calls. As a result, the bean code is not tied to a specific persistent storage mechanism (database).
- c. If the same entity beans are implemented on different J2EE servers that use different databases, we do not need to modify or recompile the bean code.

**2. Bean Managed Persistence (BMP) :**

- a. In this method, the entity bean provides an object view of the data.
- b. A Bean Managed Persistence mechanism transforms the physical data structure to a Java object.
- c. The entity bean has code that accesses the persistence environment directly.
- d. BMP can be used to reduce the overhead of CMP.

**PART-2**

*Java Database Connectivity (JDBC) : Merging Data from Multiple Tables : Joining, Manipulating, Database with JDBC, PreparedStatement, Transaction Processing, Stored Procedures.*

**CONCEPT OUTLINE : PART-2**

- JDBC stands for Java Database Connectivity.
- It provides an interface to access database.
- PreparedStatement represents the pre-compiled SQL statement.
- JDBC API is defined in the Java.sql and javax.sql package.

**Questions-Answers**

**Long Answer Type and Medium Answer Type Questions**

**Que 4.7.** What is JDBC ? How it works ?

**Answer**

1. JDBC (Java Database Connectivity) is an API which consists of various classes, interfaces, exceptions using Java.
2. JDBC is useful for both application developers and JDBC driver vendors.
3. JDBC is specially used for having connectivity with the RDBMS packages using corresponding JDBC driver.

**Working of JDBC :**

1. All Java application establishes connection with the data source and invokes classes and interfaces from JDBC driver for sending queries to the data source.
2. The JDBC driver connects to corresponding database and retrieves the result.
3. These results are based on SQL statements, which are then returned to Java applications.
4. Java application then uses the retrieved information for further processing.

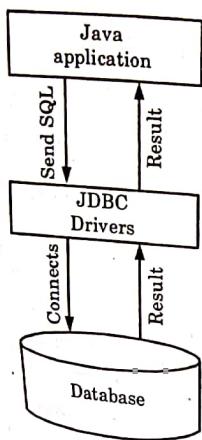


Fig. 4.7.1.

**Que 4.8.** What are the components of JDBC ?

**Answer**

#### Components of JDBC :

1. **Driver manager :**
  - a. When Java applications need connection to the database it invokes the DriverManager class.
  - b. This class then loads JDBC drivers in the memory. The driver manager also attempts to open a connection with the desired database.
2. **Connection :**
  - a. This is an interface which represents connectivity with the data source.
  - b. The connection is used for creating the statement instance.
3. **Statement :**
  - a. This interface is used for representing the SQL statements.
  - b. Some SQL statements are :  
SELECT \* FROM students\_table;  
UPDATE students\_table set name = 'Nitin' WHERE roll\_no = '1';  
c. There are two specialised statement types : PreparedStatement and callableStatement.

#### 4. ResultSet :

- a. This interface is used to represent the database resultSet.
  - b. After using SELECT SQL statement, the information obtained from the database can be displayed using ResultSet.
5. **SQL exception :** For handling SQL exceptions, this interface is used.

**Que 4.9.** Explain JDBC application architecture.

**Answer**

#### JDBC architecture :

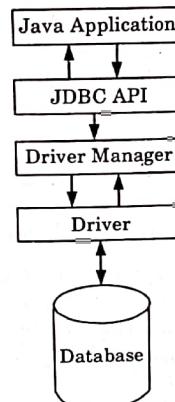


Fig. 4.9.1.

**Java application :** It is a standalone Java program, a servlet, a JSP, a Java Bean or an EJB, which uses the JDBC API to get connected and perform operations on the database data.

**JDBC API :** It is a set of classes and interfaces used in a Java program for database operations. Java.sql & Javax.sql packages provide the necessary library support.

**Driver manager :** Java program uses DriverManager class to get the connection with the database.

**Driver :** It is the software that establishes connection with the database. It is the translation software that translates the JDBC method calls. This software enables the communication between Java program and the database.

**Database :** It is a collection of all enterprise data.

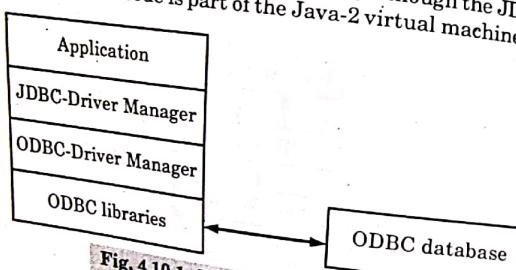
**Que 4.10.** Explain the types of JDBC drivers.

**Answer**

Types of JDBC drivers are :

**1. JDBC-ODBC bridge driver (Type 1 driver) :**

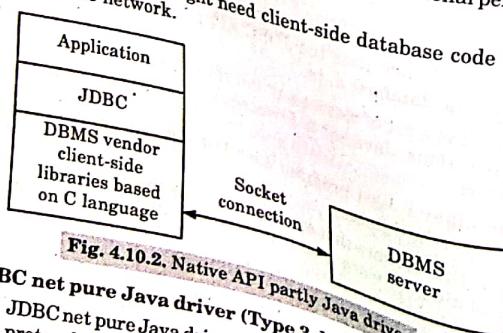
- These drivers are the bridge drivers such as JDBC-ODBC bridge.
- These drivers rely on an intermediary such as ODBC to transfer the SQL calls to the database.
- Bridge drivers often rely on native code, although the JDBC-ODBC library native code is part of the Java-2 virtual machine.



**Fig. 4.10.1. JDBC-ODBC bridge driver.**

**2. Native API partly Java driver (Type 2 driver) :**

- A native API is partly a Java driver. It uses native C language library calls to translate JDBC to native client library.
- These drivers are available for Oracle, Sybase, DB2 and other client library based RDBMS.
- Type 2 drivers use native code and require additional permission to work in an Applet.
- A Type 2 driver might need client-side database code to connect over the network.



**Fig. 4.10.2. Native API partly Java driver.**

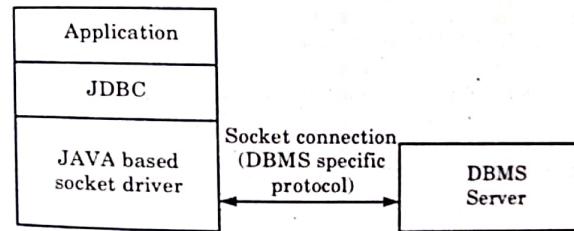
**JDBC net pure Java driver (Type 3 driver) :**

- JDBC net pure Java driver consists of JDBC and DBMS independent protocol driver.

- Here the calls are translated and sent to middle tier server through the socket.
- The middle tier contacts the database.
- Type 3 drivers call the database API on the server.

**4. Native protocol pure Java driver (Type 4 driver) :**

- A native protocol Java driver contains JDBC calls that are converted directly to the network protocol used by the DBMS server.
- The highest level of driver implements the database network API in the Java language.



**Fig. 4.10.3. Native protocol pure Java driver.**

- Type 4 drivers can also be used on thin clients as they have no native code.
- The network protocol is defined by the vendor and is typically proprietary, the driver usually comes only from the database vendor.

**Que 4.11.** Explain JDBC application architecture. What are the various types of JDBC drivers ? Write steps to connect database with the web application using JDBC. **AKTU 2015-16, Marks 15**

**Answer**

**JDBC application architecture :** Refer Q. 4.9, Page 4-11Z, Unit-4.

**Types of JDBC driver :** Refer Q. 4.10, Page 4-11Z, Unit-4.

**Steps to connect database with web application using JDBC :**

**Step 1 :** Create a database using some suitable database management package.

**Step 2 :** Initiate object for JDBC driver using following statement :

`Class.forName("com.mysql.jdbc.Driver").newInstance();`

**Step 3 :** Using DriverManager class and getConnection method we get connected to the database.

To get connected with MySQL database we use following statement :

`DriverManager.getConnection("jdbc:mysql://localhost:3306/students", "root", "system");`

**Que 4.12.** Write a Java program to retrieve data from multiple tables.

**Answer**

```
import java.sql.*;
public class jdbcConn
{
    public static void main(String[] args) throws Exception
    {
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        Connection con = DriverManager.getConnection(
            "jdbc:derby://localhost:1527/testDb", "username", "password");
        Statement stmt = con.createStatement();
        String query = "SELECT fname, lname, isbn from author inner join
        books on author.AUTHORID = books.AUTHORID";
        ResultSet rs = stmt.executeQuery(query);
        System.out.println("Fname Lname ISBN");
        while (rs.next())
        {
            String fname = rs.getString("fname");
            String lname = rs.getString("lname");
            int isbn = rs.getInt("isbn");
            System.out.println(fname + " " + lname + " " + isbn);
        }
        System.out.println();
        System.out.println();
    }
}
```

**Output:**

```
Fname Lname ISBN
Jatin Garg 123
Pankaj Sharma 113
Pankaj Sharma 112
Pankaj Sharma 122
```

**Que 4.13.** Explain PreparedStatement interface in JDBC.

**Answer**

1. The PreparedStatement interfaces define the methods and properties that enable us to send SQL or PL/SQL commands and receive data from our database.
2. They also define methods that help bridge data type differences between Java and SQL data types used in a database.

**Web Technology**

3. The PreparedStatement interface extends the statement interface, which gives us added functionality with a couple of advantages over a generic statement object.
4. This statement gives us the flexibility for supplying arguments dynamically.
5. Syntax to create PreparedStatement object:

```
PreparedStatement pstmt = null;
try {
    String SQL = "Update Employees SET age = ? WHERE id = ?";
    pstmt = conn.prepareStatement(SQL);
    ...
}
catch (SQLException e) {
    ...
}
finally {
    pstmt.close();
}
```
6. All parameters in JDBC are represented by the ? symbol, which is known as the parameter marker. We must supply values for every parameter before executing the SQL statement.
7. To close the PreparedStatement object a simple call to the close() method is made. If we close the connection object first, it will close the PreparedStatement object as well.

**Que 4.14.** Explain the steps to connect a Java application to database.

**Answer**

Steps to connect a Java application to database :

1. **Register the driver :**  
Class.forName() is used to load the driver class explicitly.  
**Example :**  
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
2. **Create a connection :**
  - a. getConnection() method of DriverManager class is used to create a connection.
  - b. Syntax :  
getConnection(String url)

```
getConnection(String url, String username, String password)
getConnection(String url, Properties info)
```

**Example :** Establishing connection with Oracle Driver  
 Connection con = DriverManager.getConnection

```
("jdbc:oracle:thin:@localhost:1521:XE", "username", "password");
```

### 3. Create SQL statement :

- createStatement() method is invoked on current connection object to create a SQL Statement.
- Syntax :

```
public Statement createStatement() throws SQLException
```

**Example :**

```
Statement s=con.createStatement();
```

### 4. Execute SQL statement :

- executeQuery() method of statement interface is used to execute SQL statements.
- Syntax :

```
public ResultSet executeQuery(String query) throws SQLException
```

**Example :**

```
ResultSet rs=s.executeQuery("select * from user");
```

```
while(rs.next())
```

```
{
```

```
System.out.println(rs.getString(1)+" "+rs.getString(2));
```

### 5. Closing the connection :

- After executing SQL statement, to close the connection and release the session.
- The close() method of connection interface is used to close the connection.
- Syntax :

```
public void close() throws SQLException
```

**Example :**

```
con.close();
```

**Que 4.15.** Write a short note on stored procedure in Java.

### Answer

- A program which contains  $n$  number of SQL statements and residing a database environment is known as stored procedure.
- Stored procedures are divided into two types :

### a. Procedure :

- A procedure is one which contains block of statements which will return either zero or more than one value.
- Syntax for creating a procedure :

```
create procedure <procedure name> (parameters)
as/is
```

```
local variables;
```

```
begin
```

```
block of statements;
```

```
end;
```

```
/
```

### b. Function :

- A function is one which contains  $n$  number of block of statements to perform some operation and it returns a single value only.

- Syntax for creating a function :

```
create function (a in number, b in number) return <return
type>
as/is
n1 out number;
begin
n1:=a+b;
return (n1);
end;
//
```

### Que 4.16

Explain the steps to execute stored procedure from JDBC.

### Answer

Steps to execute the stored procedures from JDBC are :

**Step 1 :** Create an object of CallableStatement by using the following method :  
`java.sql.Connection public CallableStatement prepareCall (String);`  
 Here, String represents a call for calling a stored procedure from database environment.

**Step 2 :** Prepare a call either for a function or for a procedure which is residing in database.

Syntax :

```
CallableStatement cs=con.prepareCall("? = call <name of the function> (?, ?, ?)");
CallableStatement cs=con.prepareCall ("call <name of the procedure> (?, ?, ?)");
```

The positional parameters numbering will always from left to right starting from 1.

**Step 3:** Specify which input parameters are using the following generalized method :

```
Public void setXXX (int, XXX);
```

**Step 4:** Specify which output parameters are using the following generalized method :

```
java.sql.CallableStatement public void registerOutParameter (int, jdbc  
database data type);
```

All the data members which are available in Types class are belongs to public static final data members.

**Step 5:** Execute the stored procedure by using the following method :

```
java.sql.CallableStatement public boolean execute ();
```

**Step 6:** Get the values out of parameters by using the following method :

```
public XXX getXXX (int);
```

Here, int represents position of out parameter. XXX represents fundamental data type or string or date.

**Que 4.17.** Write the difference between stored procedure and functions.

**Answer**

The differences between stored procedures and functions are as follows :

| S.No. | Stored Procedure   | Function   |
|-------|--|--|
| 1.    | It is used to perform business logic.                                      | It is used to perform calculation.   |
| 2.    | It must not have the return type.  | It must have the return type.  |
| 3.    | It may return more than one values.  | It may return only one value.  |
| 4.    | We can call functions from the procedure.                                  | Procedure cannot be called from function.  |
| 5.    | Procedure supports input and output parameters.                            | Function supports only input parameter.  |
| 6.    | Exception handling using try/catch block can be used in stored procedures. | Exception handling using try/catch block cannot be used in user-defined functions. |

**Que 4.18.** Explain transaction management in JDBC. What are the types of transaction ?

**Answer**

1. A transaction is a group of operation used to perform single task.
2. If all operations in the group are successful then the task is finished and the transaction is successfully completed.
3. If any one operation in the group is failed then the task is failed and the transaction is failed.

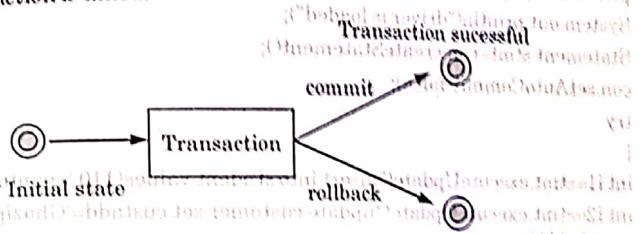


Fig. 4.18.1.

**Types of transaction :**

**1. Local transaction :** A local transaction means that all operations in a transaction are executed against one database.

**For example :** If we transfer money from first account to second account and both accounts belongs to same bank then transaction is local transaction.

**2. Global transaction :** A global transaction means that all operations in a transaction are executed against multiple databases.

**For example :** If we transfer money from first account to second account belongs to different banks then the transaction is a global transaction.

**Que 4.19.** Describe the transaction management method in JDBC with example.

**Answer**

In JDBC, Connection interface provides methods to manage transaction.

| Method                             | Description  |
|------------------------------------|--|
| void setAutoCommit(boolean status) | transaction is committed by default. It loads if not false. To turn off otherwise. |
| void commit()                      | commits the transaction.   |
| void rollback()                    | cancels the transaction.   |

**Example :**

```
import java.sql.*;
class TrxaExample
{
    public static void main(String[] args) throws Exception
    {
        Class.forName("oracle.jdbc.OracleDriver");
        Connection con = DriverManager.getConnection("jdbc:oracle:thin:@quantum-
pc:1521:xe", "system", "system");
        System.out.println("driver is loaded");
        Statement stmt = con.createStatement();
        con.setAutoCommit(false);
        try
        {
            int i1 = stmt.executeUpdate("insert into student values(110,'quantum',685)");
            int i2 = stmt.executeUpdate("update customer set custadd='Ghazipur' where
custid=111");
            int i3 = stmt.executeUpdate("delete from student where sid=101");
            con.commit();
            System.out.println("Transaction is successful");
        } //end of try
        catch (Exception e)
        {
            try
            {
                con.rollback();
                System.out.println("Transaction is failed");
            }
            catch (Exception ex)
            {
                System.out.println(ex);
            }
        } //end of catch
        stmt.close();
        con.close();
        System.out.println("connection is closed");
    } //end of main
} //end of class.
```

## Questions-Answers

### Long Answer Type and Medium Answer Type Questions

**Que 5.1. What is servlet ? Explain its life cycle. Give its characteristics.**

**OR**

**Explain the life cycle of servlet. Also write a servlet for displaying a string "HELLO WORLD!"**

**AKTU 2017-18, Marks 10**

#### Answer

**Servlet :**

1. Servlets are simple Java programs that run on the servers.
2. Servlets are most commonly used with HTTP. So, servlets are also called as HTTP servlet.
3. Servlet can process and store the data submitted by an HTML form.
4. Servlets are useful for providing the dynamic contents.
5. Servlets can be used in the cookies and session tracking.

**The life cycle of a servlet is determined by three method :**

**a. The init() method :**

- a. The init( ) method is called by the servlet container after the servlet class has been initiated.
- b. The servlet container calls this method exactly once to indicate that the servlet is being placed into service. The init( ) method must complete successfully before the servlet can receive any requests.

- c. The syntax of this method is as follows :
- ```
public void init(ServletConfig config) throws ServletException
```
2. The service() method :
- The service( ) method is called by the servlet container after the init( ) method to allow the servlet to respond to a request.
  - The syntax of this method is as follows :
- ```
public void service(ServletRequest request, ServletResponse response) throws ServletException, java.io.IOException
```
- c. The servlet container passes a object which contains the client request and the servlet response.
3. The destroy() method :
- The servlet container calls the destroy( ) method before removing a servlet instance from service.
  - The destroy method is used to clean up the resources that are being held (for example, memory, file handles, and threads).
  - The syntax of this method is as follows :
- ```
public void destroy()
```

**Characteristics of servlet :**

- Servlet operates on input data that is encapsulated in a request object.
- Servlet responds to a query with data encapsulated in a response object.
- Servlet can call EJB components to perform business logic functions.
- Servlet can call JSPs to perform page layout functions.
- Servlet is extensible.
- Servlet can call other servlets.

**Servlet for displaying "HELLO WORLD" :**

```
// Import required java libraries
import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
// Extend HttpServlet class
public class HelloWorld extends HttpServlet {
    private String message;
    public void init() throws ServletException {
        // Do required initialization
        message = "HELLO WORLD";
    }
    public void doGet(HttpServletRequest request,
```

```
HttpServletResponse response)
throws ServletException, IOException
```

```
{
    // Set response content type
    response.setContentType("text/html");
    // Actual logic goes here.
    PrintWriter out = response.getWriter();
    out.println("<h1>" + message + "</h1>");
}
```

```
public void destroy() {
```

```
// do nothing.
```

```
}
```

```
}
```

**Que 5.2.** What do you mean by Common Gateway Interface (CGI) ?

How does CGI work ?

AKTU 2014-15, Marks 06

**Answer**

- CGI is an acronym for the Common Gateway Interface which is a standard protocol for running programs within a web server.
- The CGI protocol allows external program to interface with programs such as database management software and to access the networking facilities provided by the HTTP server software.
- CGI programs are dynamic and the state of their variables alters as they execute.
- CGI allows a WWW server to provide information to WWW clients that would otherwise not be available to those clients.

**Working of CGI :**

- A reader sends a URL that causes the AOLserver to use CGI to run a program.
- The AOLserver passes input from the reader to the program and output from the program back to the reader. CGI acts as a "gateway" between the AOLserver and the program.
- The program run by CGI can be any type of executable file on the server platform. For example, we can use C, C++, Perl, Unix shell scripts.
- Since CGI is a standard interface used by many web servers, there are lots of example programs and function libraries available on the World Wide Web and by FTP.

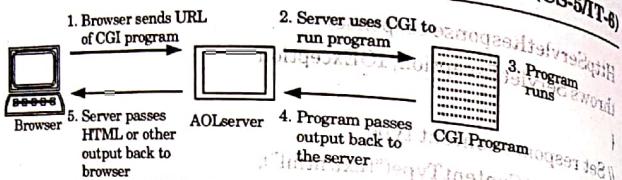


Fig. 5.2.1. Working of CGI.

**Que 5.3.** What is servlet? Explain its life cycle. Illustrate some characteristics of servlet. How does servlet score over CGI?

AKTU 2015-16, Marks 10

**Answer**

**Servlet, its lifecycle and characteristics of servlet :** Refer Q. 5.1, Page 5-2Z, Unit-5.

- Java servlets have following advantage over CGI and other APIs:**
- Platform independence :** Java servlets are pure Java program, so it's platform independent. It can run on any servlet enabled web server.
  - Performance :** In case of servlets, initialization takes place very first time it receives a request and remains in memory till times out or server shut downs. This helps us to develop high speed data driven websites.
  - Extensibility :** Java servlets are developed in Java which is robust, well-designed and object-oriented language which can be extended into new objects.
  - Safety :** Java servlet provides a very good safety features like memory management, exception handling features and emerged as a very powerful web server extension.
  - Secure :** Servlets are server-side components, so it inherits the security provided by the web server.

**Que 5.4.** Explain servlet architecture.

**Answer**

**Servlet architecture includes :**

Servlet interface

↓

Generic servlet

↓

HttpServlet

↓

MyServlet

**1. Servlet interface :**

- To write a servlet we need to implement **Servlet interface**.
- Servlet interface can be implemented directly or indirectly by extending **GenericServlet** or **HttpServlet** class.

**2. GenericServlet :**

- GenericServlet provides simple versions of the lifecycle methods **init** and **destroy** and of the methods in the **ServletConfig** interface.
- It also implements the **log** method which is declared in the **ServletContext** interface.

**3. HttpServlet :**

- HttpServlet is an abstract class present in **javax.servlet.http** package and has no abstract methods.
- It extends **GenericServlet** class.
- When the servlet container uses HTTP protocol to send request, then it creates **HttpServletRequest** and **HttpServletResponse** objects.

**4. MyServlet :** MyServlet is the combination of servlet interface, genericServlet and HttpServlet.

**Que 5.5.** Write short notes on servlet interface.

**Answer**

- Servlet interface provides common behaviour to all the servlets.
- Servlet interface needs to be implemented for creating any servlet (either directly or indirectly).
- Servlet interface defines methods that are implemented by all servlets.
- Following are five methods defined in servlet interface:

S.No.	Method	Description
1.	public void init(ServletConfig config)	It initializes the servlet. It is the life cycle method of servlet and invoked by the web container only once.
2.	public void service(ServletRequest request, ServletResponse response)	It provides response for the incoming request. It is invoked at each request by the web container.
3.	public void destroy()	It is invoked only once and indicates that servlet is being destroyed.
4.	public ServletConfig getServletConfig()	It returns the object of <b>ServletConfig</b> .
5.	public String getServletInfo()	It returns information about servlet such as writer, copyright, version etc

**Que 5.6.** Explain how servlet works.**Answer**

- When the web server starts, the servlet container deploys and loads all the servlets.
- Servlet container creates ServletContext object which act as an interface and defines the set of methods that a servlet can use to communicate with the servlet container.
- Once the servlet is loaded, the servlet container creates the instance of servlet class. For each instantiated servlet, its init() method is invoked.
- Client (user browser) sends an Http request to web server on a certain port.
- Then, the servlet container creates HttpServletRequest and HttpServletResponse objects.
- The HttpServletRequest object provides the access to the request information and the HttpServletResponse object allows us to change the http response before sending it to the client.
- The servlet container produce a new thread that calls service() method for each client request.
- The service() method dispatches the request to the correct handler method based on the type of request.

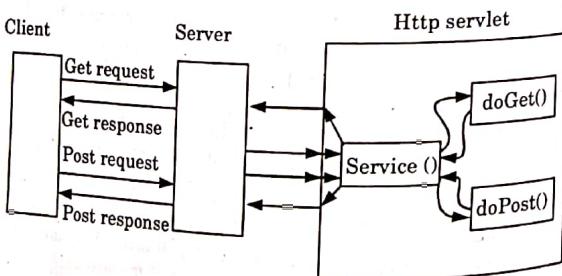


Fig. 5.6.1.

8. When servlet container shuts down, it unloads all the servlets and calls destroy() method for each initialized servlets.

**Que 5.7.** Explain handling in HTTP get and post request in servlet with example.**Answer**

- To handle HTTP requests in a servlet, extend the HttpServlet class and override the servlet methods that handle the HTTP requests that our servlet supports.
- The methods that handle these requests are doGet and doPost.

**Handling Get requests :**

- Handling Get requests involves overriding the doGetmethod.

**For example :**

```

public class BookDetailServlet extends HttpServlet {
    public void doGet (HttpServletRequest request,
                      HttpServletResponse response)
                      throws ServletException, IOException
    {
        ...
        // set content-type header before accessing the Writer
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        // then write the response
        out.println("<html>" +
                   "<head><title>Book Description</title></head>" +
                   ...
                   );
        //Get the identifier of the book to display
        String bookId = request.getParameter("bookId");
        if (bookId != null) {
            // and the information about the book and print it
            ...
        }
        out.println("</body></html>");
        out.close();
    }
    ...
}

```

- The servlet extends the HttpServlet class and overrides the doGetmethod.
- Within the doGetmethod, the getParametermethod gets the servlet's expected argument.
- To respond to the client, the example doGetmethod uses a Writer from the HttpServletResponse object to return text data to the client. Before accessing the writer, the example sets the content-type header.

5. At the end of the `doGetmethod`, after the response has been sent, the Writer is closed.

#### Handling Post requests :

1. Handling Post requests involves overriding the `doPostmethod`.

For example :

```
public class ReceiptServlet extends HttpServlet {
    public void doPost(HttpServletRequest request,
                      HttpServletResponse response)
        throws ServletException, IOException {
        // set content type header before accessing the Writer
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        // then write the response
        out.println("<html>" +
                   "<head><title> Receipt </title>" +
                   "</head><body> Thank you for purchasing your books from us" +
                   " " + request.getParameter("cardname") + "</body>" +
               "</html>");
        out.close();
    }
}
```

2. The servlet extends the `HttpServlet` class and overrides the `doPostmethod`.  
 3. Within the `doPostmethod`, the `getParametermethod` gets the servlet's expected argument.  
 4. To respond to the client, the example `doPostmethod` uses a `Writer` from the `HttpServletResponse` object to return text data to the client. Before accessing the writer the example sets the content-type header.  
 5. At the end of the `doPostmethod`, after the response has been set, the Writer is closed.

#### Que 5.8. Write short notes on session tracking.

##### Answer

1. Session tracking is a way to maintain state (data) of a user. It is also known as session management in servlet.

2. It is used to recognize the particular user, so to recognize user.  
 3. Each time user requests to the server, server treats the request as the new request and maintain the state of an user using session tracking techniques to recognize a particular user.  
 4. HTTP is stateless that means each request is considered as the new request. It is shown in the Fig. 5.8.1.

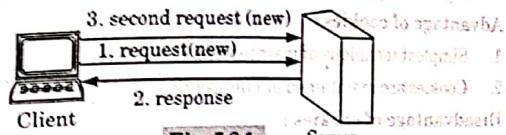


Fig. 5.8.1.

5. There are four techniques used in session tracking:

- Cookies
- Hidden Form Field
- URL Rewriting
- HttpSession

#### Que 5.9. Explain cookies with its advantage and disadvantage. Also give its type.

##### Answer

- Cookie is a key value pair of information sent by the server to the browser which is saved in the client computer.
- A cookie is a small piece of information that is persisted between the multiple client requests.
- A cookie has a name, a single value, and optional attributes such as a comment, path and domain qualifiers, a maximum age, and a version number.

#### Working of cookie :

- Client sends a request to the server.
- Cookie is added with request from the servlet and stored in the cache of the browser.
- After that if response is sent by the server, cookie is added with it by default.

- Server can identify the client using the cookie.

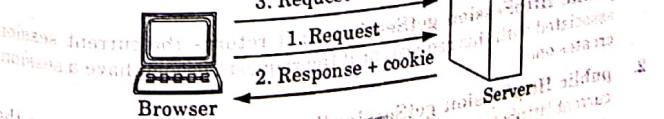


Fig. 5.9.1.

**There are two types of cookies in servlet :**

1. Non-persistent cookie : It is valid for single session only. It is removed each time when user closes the browser.
2. Persistent cookie : It is valid for multiple sessions. It is not removed each time when user closes the browser. It is removed only if user logout or signout.

**Advantage of cookies :**

1. Simplest technique of maintaining the state.
2. Cookies are maintained at client-side.

**Disadvantage of cookies :**

1. It will not work if cookie is disabled from the browser.
2. Only textual information can be set in cookie object.

**Que 5.10.** Write short notes on session tracking with HttpSession.

#### Answer

1. The HttpSession object is used for session tracking.
2. A session contains information specific to a particular user across the whole application.
3. When a user enters into a website (or an online application) for the first time HttpSession is obtained via request.getSession(), the user is given a unique ID to identify his session. This unique ID can be stored into a cookie or in a request parameter.

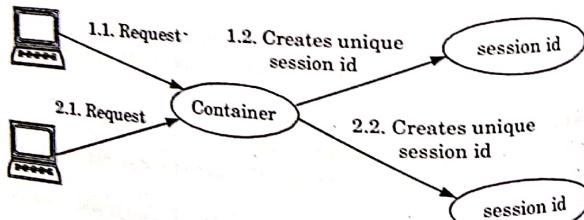


Fig. 5.10.1.

The HttpServletRequest interface provides two methods to get the object of HttpSession :

1. public HttpSession getSession() : It returns the current session associated with this request, or if the request does not have a session, creates one.
2. public HttpSession getSession(boolean create) : It returns the current HttpSession associated with this request or, if there is no current session and create is true, returns a new session.

**Commonly used methods of HttpSession interface are :**

1. public String getId() : It returns a string containing the unique identifier value.
2. public long getCreationTime() : It returns the time when this session was created, measured in milliseconds.
3. public long getLastAccessedTime() : It returns the last time the client sent a request associated with this session.
4. public void invalidate() : It invalidates this session then unbinds any objects bound to it.

**Que 5.11.** Create a form in HTML taking account number from user as input then write a servlet program receiving this from data and connect it with database by using JDBC. Then send the current account balance of user stored in specific database back to user as response. Also, mention all the assumed required data like table name, database name and fields name etc.

**AKTU 2017-18, Marks 10**

#### Answer

Let us assume a table "account\_detail" which contains the details about the Account number.

```

create table account_detail
(
    ac_no number,
    name VARCHAR2(40),
    address VARCHAR2(40),
    balance number,
    CONSTRAINT "account_detail_pk" PRIMARY KEY ("ac_no") ENABLE
)
  
```

#### Index.html :

This page get the account number from the user and forwards this data to servlet which is responsible to show the records based on the given account number.

```

<html>
<body>
<form action="servlet/Search">
Enter A/c No.:<input type="text" name="ac"/><br>
<input type="submit" value="search"/>
</form>
  
```

```

</body>
</html>

```

**Search.java :**

This is the servlet file which gets the input from the user and maps this data with the database and prints the record for the matched data. In this page, we are displaying the column name of the database along with data. So, we are using ResultSetMetaData interface.

```

import java.io.*;
import java.sql.*;
import javax.servlet.ServletException;
import javax.servlet.http.*;
public class Search extends HttpServlet {
    public void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();
        String rollno=request.getParameter("ac");
        int ac=Integer.valueOf(ac_no);
        try{
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con=DriverManager.getConnection(
                "jdbc:oracle:thin:@localhost:1521:xe","system","oracle");
            ps=con.prepareStatement("select * from result where ac_no=?");
            ps.setInt(1,ac);
            out.print("<table width=50% border=1>");
            out.print("<caption>Result:</caption>");
            ResultSet rs=ps.executeQuery();
            /* Printing column names */
            ResultSetMetaData rsmd=rs.getMetaData();
            int total=rsmd.getColumnCount();
            out.print("<tr>");
            for(int i=1;i<total;i++){
                out.print("<th>" + rsmd.getColumnName(i) + "</th>");
            }
            out.print("</tr>");
            /* Printing result */
            while(rs.next()){

```

```

out.print("<tr><td>" + rs.getInt(1) + "</td><td>" + rs.getString(2) +
"</td><td>" + rs.getString(3) + "</td><td>" + rs.getString(4) +
"</td></tr>");
out.print("</table>");
} catch (Exception e2) {e2.printStackTrace();}
finally{out.close();}
}
}

```

#### web.xml file :

This is the configuration file which provides information of the servlet to the container.

```

<web-app>
<servlet>
<servlet-name>Search</servlet-name>
<servlet-class>Search</servlet-class>
</servlet>
<servlet-mapping>
<servlet-name>Search</servlet-name>
<url-pattern>/servlet/Search</url-pattern>
</servlet-mapping>
</web-app>

```

#### PART-2

**Java Server Page (JSP) : Introduction, Java Server Page Overview, A First Java Server Page Example, Implicit Objects, Scripting, Standard Actions, Directives, Custom Tag Libraries.**

#### CONCEPT OUTLINE : PART-2

- Java Server Page (JSP) allows us to separate dynamic content from web pages.
- JSP page is a combination of different protocols, components and formats.
- JSP directives are JSP elements which provide global information about an entire JSP page.
- Tomcat server helps in implementation of Java servlet and JSP technology.

#### Questions-Answers

Long Answer Type and Medium Answer Type Questions

**Que 5.12.** What do you mean by JSP? Explain the architecture of JSP. How JSP provides better performance?

AKTU 2013-14, Marks 10

**Answer**

- Java Server Pages (JSP) is a technology that helps software developers to create dynamically generated web pages based on HTML, XML or other document types.
- JSP supports both scripting and element based dynamic content, and allows developers to create their own tag libraries.
- JSP pages are compiled for efficient server processing.
- JSP is platform independent. So, it can be easily upgraded or switched without affecting JSP based applications.
- JSP tags are reusable. So, the pages run faster.

**JSP architecture :**

- Java Server Pages are part of a 3-tier architecture.
- A server (application or web server) supports the Java Server Pages.
- This server will act as a mediator between the client browser and database.
- The following diagram shows the JSP architecture :

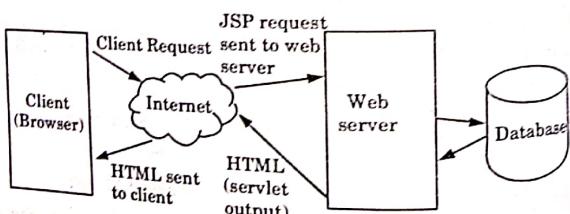


Fig. 5.12.1. JSP architecture.

- The user goes to a JSP page and makes the request via internet in web browser.
- The JSP request is sent to the web server.
- Web server accepts the requested .jsp file and passes the JSP file to the JSP Servlet Engine.
- If the JSP file has been called the first time then the JSP file is parsed otherwise servlet is instantiated. The next step is to generate a servlet from the JSP file. The generated servlet output is sent via the Internet from web server to user's web browser.

- Now in last step, HTML results are displayed on the user's web browser.

**JSP provides better performance :** JSP allows to embed the Java code directly in the HTML file and generates the contents dynamically. This helps to enhance the performance of JSP.

**Que 5.13.** What is JSP? What are the advantages of JSP over various server-side programs?

AKTU 2014-15, Marks 3.5

**Answer**

JSP : Refer Q. 5.12, Page 5-15Z, Unit-5.

**Advantages of JSP over other server-side programs are :**

- JSP is platform independent whereas other server-side programs are not.
- JSP code is portable but other server-side programs are not.
- JSP can be used for large application but other server-side programs are used for small application.
- JSP uses static type checking while other uses dynamic type checking.

**Que 5.14.** Explain JSP processing.

**Answer**

- In JSP processing, the JSP engine compiles the servlets upto an executable class and forwards the original request to the servlet engine and execute it on the web server.
- JSP pages can be processed using JSP container only.
- Following are the steps that need to be followed while processing the request for JSP page :
  - Client makes a request for required JSP page to the server.
  - The server must have JSP container so that JSP request can be processed.
  - On receiving request the JSP container searches and then reads the desired JSP page.
  - Then JSP page is converted to corresponding servlet. Basically any JSP page is a combination of template text and JSP element.
  - Every template text is translated into corresponding println statement.

**Que 5.15.** How error handling and debugging is done in JSP? Explain with suitable example.

AKTU 2014-15, Marks 3.5

**Answer**  
Error handling in JSP can be done as follows :

a. Error handling within the current page :

1. The JSP page can either catch exceptions that are thrown by the data bean or it can check for error codes that are set within each data bean, depending on how the data bean was activated.
2. The JSP page can then take an appropriate recovery action that is based on the error received.
3. The JSP page can use try and catch blocks to capture the exception so that it can take appropriate action that is based on the exception type.

For example : JSP uses try and catch Java statements :

```
SomeDataBean sdb = new SomeDataBean();
sdb.setSomeProperty(" ");
try {
    com.ibm.commerce.beans.DataBeanManager.activate(sdb,
request);
    catch(Exception ex){
        //Handle the exception in whichever way we want.
    }
}
```

b. Error handling at page level :

1. A JSP page can specify its own default error JSP page from an exception that is occurring within it, through the JSP error tag.
2. This enables a JSP page to specify its own handling of an error.
3. A JSP page that does not contain a JSP error tag has an error fall through to the application-level error JSP page.

For example :

- i. Create a single error JSP page that handles the errors that occur across all the other JSP pages in the application. To specify a JSP page as an errorHandler page, use this JSP page directive :

```
<%@ page isErrorPage="true" %>
```

In the errorHandler JSP page, use ErrorDataBean or StoreErrorDataBean to retrieve more information about the exception and display messages.

- ii. Include the errorHandler JSP page in other JSP pages, by using this JSP directive to specify that if exceptions occur on the current page, forward the request to errorHandler.jsp :

```
<%@ page errorPage="/errorHandler.jsp" %>
```

**Debugging in JSP :** Debugging is the process to trace the error in the application. Since, JSP applications are client interactive. Hence, errors are difficult to reproduce.

**JSP debugging can be done :**

1. By using System.out.println() :

- a. System.out.println() is easy to use as a marker to test whether a certain piece of code is being executed or not.
- b. We can print out variable values as well.
- c. Syntax to use System.out.println() is :

```
System.out.println("Debugging message");
```

2. By using JDB debugger :

- a. We can debug JSP and servlets with the same jdb commands we use to debug an applet or an application.
- b. To debug a JSP or servlet, we can debug sun.servlet.http.HttpServer, then observe as HttpServer executes the JSP/servlets in response to HTTP requests we make from a browser.

3. By setting classpath :

- a. Set our debugger's classpath. This helps us to find sun.servlet.http.HttpServer and the associated classes.
- b. This helps us to find our JSP and support classes, typically ROOT\WEB-INF\classes
- c. Once we have set the proper classpath, start debugging sun.servlet.http.HttpServer.

4. By using comments :

- a. The JSP uses Java comments, single line (// ...) and multiple line /\* ... \*/ comments can be used to temporarily remove parts of our Java code.
- b. If the bug disappears, find out the problem in the comment.

**Que 5.16.** Write note on Tomcat server. **AKTU 2015-16, Marks 05**

OR

Explain in detail the Tomcat server.

**AKTU 2016-17, Marks 10**

**Answer**

1. Tomcat server is a Java-capable HTTP server, which could execute special Java programs known as "Java servlet" and "Java Server Pages".
2. Tomcat can be used standalone, but it is often used "behind" traditional web servers such as Apache.
3. Tomcat server runs on a specific TCP port from a specific IP address.

4. The default TCP port number for HTTP protocol is 80, which is used for the production HTTP server.  
 5. To test HTTP server, we can choose any unused port number between 1024 and 65535.

**Que 5.17.** What are the steps for running JSP program in Tomcat server?

AKTU 2014-15, Marks 3.5

**Answer**

Steps for running JSP program in Tomcat server are :

1. Create a JSP code and save it using the filename extension.jsp.
2. Copy this JSP file to the directory named webapps. This directory is present within the Tomcat directory.
3. Start the Tomcat server by typing the command startup.
4. Open some suitable web browser type the path for JSP code with the prefix http://localhost. Localhost is the default DNS for Tomcat web server.

**Que 5.18.** Write a simple JSP page for displaying the message : "This is my first JSP page !!!".

**Answer**

```
<%@ page language="java" contentType="text/html; charset=US-ASCII"
<%@ page import="java.util.*">
<html>
  <!-- This is basic JSP page -->
  <title>JSP Demo</title>
  <body>
    <%--Displaying the message on the browser --%>
    <% out.println("This my first JSP page!!!"); %>
  </body>
</html>
```

**Que 5.19.** What are the implicit objects available to JSP page? Explain.

AKTU 2014-15, Marks 3.5

OR

Explain implicit objects available in JSP with example.

AKTU 2017-18, Marks 10

Following are the different type of implicit objects which are available to JSP page :

1. **Application object :** The application object has an application scope and contains a reference to the instance of a class that implements the javax.servlet.ServletContext interface that represents the application.
  2. **Config object :**
    - a. The config object has a page scope. This object implements the javax.servlet.ServletConfig interface.
    - b. The config object gives access to configuration data for initializing the JSP.
  3. **Session object :**
    - a. HttpSession class represents the current session of the JSP page.
    - b. It represents the scope of this session, and it is useful in order to keep attributes and values and providing them in different JSP pages of same application.
  4. **Out object :**
    - a. The out object also has a page scope. Out object is an instance of javax.servlet.jsp.JspWriter class.
    - b. By using this object the text is added to the response message body.
  5. **Page object :**
    - a. Page object is an instance of java.lang.Object class.
    - b. The page object is a reference to the current instance of the JSP.
- Example :**
- ```
<%@ page language="java" contentType="text/html; charset=US-ASCII"
pageEncoding="US-ASCII"%>
<%@ page import="java.util.Date" %>
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=US-ASCII">
  </head>
  <title>Index JSP Page</title>
  </head>
  <body>
    <%-- out object example --%>
    <h4>Hi There</h4>
    <strong>Current Time is</strong>: <% out.print(new Date()); %><br><br>
    <%-- config object example --%>
    <strong>User init param value</strong>
    </strong><%=config.getInitParameter("User")%><br><br>
    <%-- application object example --%>
    <strong>User context param value</strong>;
    </strong><%=application.getInitParameter("User")%><br><br>
    <%-- session object example --%>
    <strong>User Session ID</strong>;<%=session.getId()%><br><br>
    <%-- page object example --%>
    <strong>Generated Servlet Name</strong>;
```

```
<%page.getClass().getName()%>
<html>
<body>
```

**Ques 5.20.** What are the different types of standard action tags used in JSP?

AKTU 2014-15, Marks 25

OR

What are standard actions in JSP? Illustrate with example.

AKTU 2017-18, Marks 10

### Answer

Different types of standard action tags used in JSP are :

1. **<jsp:useBean> :**

- a. This action associates an instance of a Java Bean defined with a given scope and ID, through a newly declared scripting variable of the same ID.
- b. The **<jsp:useBean>** action is very flexible.
- c. Its exact semantics depends on the values of the given attributes.
- d. The attributes for the **<jsp:useBean>** are : id, scope, class, beanName, type.

2. **<jsp:setProperty> :**

- a. This action helps to integrate Java Beans into JSPs.
- b. It sets the value of a Beans property.
- c. This action has the attributes name, property, param, value.

3. **<jsp:getProperty> :**

- a. This action gets a property value from a Java Beans component and adds it to the response.
- b. Attribute of this action are : name, property.

4. **<jsp:include> :**

- a. This action provides a mechanism for including additional static and dynamic resources in the current JSP page.
- b. The attributes for this action are : page, flush.

5. **<jsp:attribute> :** This action is used to set the value of an action attribute based on the body of this element.

- a. This action is used to set the action element body based on the body of this statement.
- b. It is required when the action element body contains **<jsp:attribute>** action element.

7. **<jsp:element> :** It dynamically generates an XML element, optionally with attributes and a body defined by nested **<jsp:attribute>** and **<jsp:body>** actions.

- 8. **<jsp:text> :** This action is used to encapsulate template text that should be used in JSP pages written as XML documents.

Example of **<jsp:include>** Action :

```
<html>
<head>
<title>The include Action Example</title>
</head>
<body>
<center>
<h2>The include action Example</h2>
<jsp:include page = "date.jsp" flush = "true" />
</center>
</body>
</html>
```

Output :

The include action Example  
Today's date: 12-june-2018 14:54:22

Example of **<jsp:useBean>, <jsp:setProperty>, <jsp:getProperty>** actions :

Let us define a test bean that will further be used :

```
/* File: TestBean.java */
package action;
public class TestBean {
private String message = "No message specified";
public String getMessage() {
return message;
}
public void setMessage(String message) {
this.message = message;
}
}
```

Now use the following code in main.jsp file. This loads the bean and sets a simple String parameter.

```
<html>
<head>
<title>Using JavaBeans in JSP</title>
</head>
<body>
<center>
<h2>Using JavaBeans in JSP</h2>
<jsp:useBean id = "test" class = "action.TestBean" />
<jsp:setProperty name = "test" property = "message" value = "Hello JSP..."/>
<p>Got message....</p>
<jsp:getProperty name = "test" property = "message"/>
</center>
</body>
</html>
```

**Output :**

Using JavaBeans in JSP

Got message...  
Hello JSP...

**Example of <jsp:element>, <jsp:attribute> and <jsp:body> actions :**

```
<%@page language = "java" contentType = "text/html"%>
<html xmlns = "http://www.w3.org/1999/xhtml"
      xmlns:jsp = "http://java.sun.com/JSP/Page">
<head><title>Generate XML Element</title></head>
<body>
<jsp:element name = "xmlElement">
<jsp:attribute name = "xmlElementAttr">
Value for the attribute
</jsp:attribute>
<jsp:body>
Body for XML element
</jsp:body>
</jsp:element>
</body>
</html>
```

**Example of <jsp:text> action :**

```
<jsp:text><![CDATA[<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML
1.0 Strict//EN"
 "DTD/xhtml1-strict.dtd">]]></jsp:text>
<body>
<books><book><jsp:text>
Welcome to JSP Programming
</jsp:text></book></books>
</body>
</html>
```

**Que 5.21.** What do you mean by JSP processing? How JSP pages are handled? Explain various JSP directives with suitable examples.

What are JSP directives? Explain various types of directives with example.

**AKTU 2015-16, Marks 10**

**AKTU 2017-18, Marks 10**

**Answer**

JSP processing : Refer Q. 5.14, Page 5-16Z, Unit-5.  
JSP page handling : Refer Q. 5.15, Page 5-16Z, Unit-5.

**5-23 Z (CS-5/IT-6)****5-24 Z (CS-5/IT-6)****JSP directives :**

1. Directives are JSP elements that provide global information about an entire JSP page.
2. All directives have scope of the entire JSP file.
3. The directive elements specify information about the page that remains the same between requests.

There are three possible directives currently defined by the JSP specification :

**a. Page directive :**

1. The page directive defines information that is globally available for JSP.
2. The page directive is a JSP tag that is used in almost every JSP file and defines a number of attributes that can affect the whole page.
3. The syntax is as follows :

`<%@ page attribute%>`

**Example :**

```
<%@ page language="java" contentType="text/html; charset=ISO-
8859-1" pageEncoding="ISO-8859-1"%>
```

<%@page  
import="com.javacode.examples.jspdirectivesexample.Pizza"%>  
<%@ page import="java.util.\*"%>  
In the first line of example, language, contentType and pageEncoding page directive attributes are in the same directive statement.

**b. Include directive :**

1. The include directive is used to insert text and code at JSP translation time.
2. It includes a static file in a JSP file.
3. It has the following syntax :

`<%@ include file = "relativeURL"%>`

**Example :**  
In the example "header.html" page is inserted into the Pizza form :

```
...
<head>
<meta charset="UTF-8">
<title>Jsp Directives Example</title>
<link rel="stylesheet" href="/static/css/pizzaorder.css">
</head>
<body>
<%@ include file="header.html" %>
<form action="orderResult.jsp" method="POST">
<h3>Pizza Types</h3>
<div>
...
header.html:
header.html: Java Code Examples<h4>
<h4>Java Code Examples<h4>
```

**c. Taglib directive :**

1. The taglib directive declares that the page uses custom user defined tags, it also defines the tags library.
2. The term custom tag refers to both tags and elements.
3. A tag is simply a short piece of a markup that is a part of JSP element.
4. The syntax is as follows :

```
<%@ taglib uri="tagLibraryURI" prefix="tagPrefix"%>
```

**Example :**

Following example shows a declaration sample in a JSP page :

```
<%@ taglib prefix="jgc" uri="WEB-INF/custom.tld"%>
<%
    jgc:HelloWorld/>
%>
```

**Que 5.22.** Explain various types of JSP scripting elements with example.

**Answer**

1. JSP scripting elements allows us to insert Java code into Java Servlet generated from JSP page.
2. Following are the four scripting elements :

**a. JSP comment :**

- i. JSP comment is to document the code.
- ii. JSP comment is used to note some parts of JSP page to make it clearer and easier to maintain.
- iii. The syntax of JSP comment is as follows :

```
<%-- This is a JSP comment --%>
```

**b. Expression :**

- i. The expression is one of the most basic scripting elements in JSP.
- ii. The expression is used to insert value directly to the output.
- iii. The syntax of an expression is as follows :

```
<%= expression%>
```

**c. Scriptlet tag :**

- i. Scriptlet is similar to expression except for the equal sign "=".
- ii. We can insert any plain Java code inside a scriptlet.
- iii. The syntax of scriptlet tag is as follows :

```
<% // any java source code here %>
```

**d. Declaration tag :**

- i. We can declare static member, instance variable and methods inside declaration tag.

**ii. Syntax of declaration tag :**

```
<%! declaration %>
```

**Example :**

```
<html>
<head>
<title>My First JSP Page</title>
</head>
<%--This is declaration tag--%>
<%
int count = 0;
%>
<body>
Page Count is :
<% out.println(++count); %>
<%= new java.util.Date()%>
</body>
</html>
```

**Que 5.23.** What are Java Beans ? Why they are used ? Write a JSP page and use an existing Java Beans in JSP page by using the standard action. Write the program with describing the output ?

**AKTU 2017-18, Marks 10**

**Answer**

Java Beans and Java Beans are used because : Refer Q. 4.3, Page 4-5Z, Unit-4.

**JSP page :**

Login.jsp :

<body>

<h2>Using Java Beans with JSP</h2>

<form method="get" action="http://localhost:7001/examplesWebApp/

Receive.jsp">

Enter User Name <input type="text" name="user"> <br>

Enter Password <input type="password" name="pass"> <br>

<input type="submit">

```

</form>
</body>
Receive.jsp :
<body>
<jsp:useBean id="snr" class="pack.ValidateBean"/>
<jsp:setProperty name="snr" property="user"/>
<jsp:setProperty name="snr" property="pass"/>
You entered user name as <jsp:getProperty name="snr" property="user"/>
<br>
You entered user password as <jsp:getProperty name="snr" property="pass"/>
/> <br>
<br>
You are a <%= snr.validate("Rao", "java") %> user. <br>
<b>Thank You</b>
</body>
ValidateBean.jsp :
package pack;
public class ValidateBean {
    String user;
    String pass;
    public ValidateBean(){}}
    public void setUser(String user) {
        this.user = user;
    }
    public String getUser() {
        return user;
    }
    public void setPass(String pass) {
        this.pass = pass;
    }
    public String getPass() {
        return pass;
    }
    public String validate(String s1, String s2) {
        if(s1.equals(user) && s2.equals(pass))
            return "VALID";
        else
            return "INVALID";
    }
}

```

**Output :**

Using Java Beans with JSP

Enter User Name

Enter Password

**Submit**

You entered user name as Rao

You entered user password as java

You are a VALID user.

Thank You

**Explanation of Output :**

1. The getProperty action calls get method and gets the value of property.
2. The first statement calls getUser() method and retrieves the value of the variable user (set earlier with set method) and directly puts in output stream of client.
3. The statement calls validate() method of ValidateBean and checks the user name and password entered by the user with Rao and java. The result of validation is returned to JSP expression which sends to client.

**Que 5.24. Explain custom tag libraries with its declaration.**

**Answer**

1. A custom tag library is a collection of the Tag Library Descriptor (TLD) and all files for a related set of custom actions.
2. TLD and all files are packaged in a JAR file.
3. A custom actions can access to all information about the request and can add content to the response body.
4. Custom action is implemented as a Java class or as a tag file.
5. The name of the tag file and other information are specified in a file called Tag Library Descriptor (TLD).

**Declaration of custom tag library includes :**

- a. **uri :**
  - i. The uri attribute find the class or tag file for each custom actions.
  - ii. It contains a string container which is used to locate the TLD for the library.
- b. **prefix :**
  - i. The prefix attribute assigns a label to the tag library.
  - ii. We use this label to reference its associated tag library when writing our pages using custom JSP tags.
  - iii. Custom tag library defines a default prefix.

&lt;mytaglib:newtag&gt;

**For example :**

```
<%@ page contentType = "text/html" %>
<%@ taglib prefix = "ora" uri = "orataglib" %> //This is custom tag library
<html>
<head>
<title>Messages of the Day </title>
</head>
<body bgcolor = "white">
<h1>Messages of the Day</h1>
<h2>Deep Thoughts - by Mahatma Gandhi</h2>
<i><ora:motd category = "thoughts" /> </i>
<h2>Quotes From the Famous and the Unknown</h2>
<i><ora:motd category = "quotes" /> </i>
</body>
</html>
```

**Que 5.25.** How data can be shared between JSPs ? Explain.

**Answer**

1. JSP application consists of more than a single page, and multiple pages often need access to the same information and server-side resources.
2. We need a way to pass data from one page to another when same request is processed by multiple pages.

- Sharing session data :**
1. When a user need same data over the multiple request in a session.
  2. This type of sharing is for different requests of same user.
  3. This type of sharing of data is called sharing of session data.
  4. For example, in a travel agency, date and destination is important to remember for booking the flight.

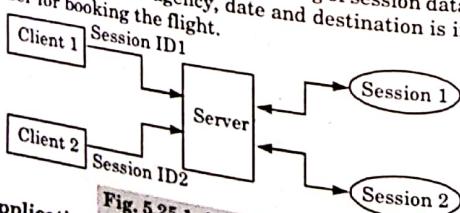


Fig. 5.25.1. Session scope.

- Sharing application data :**
1. When the different users make same request through multiple pages to an application.

2. In this case, the information is saved in the application by one page and later it can be accessed by another page, even if the two pages were requested by different users.
3. This type of sharing of data is called sharing of application data.

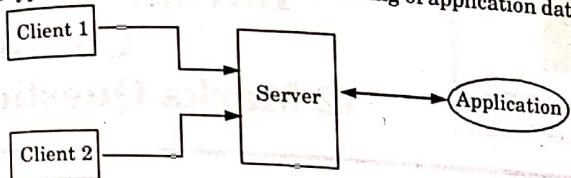


Fig. 5.25.2. Application scope.

**VERY IMPORTANT QUESTIONS**

Following questions are very important. These questions may be asked in your SESSIONALS as well as UNIVERSITY EXAMINATION.

**Q. 1. What is servlet ? Explain its life cycle.**

**Ans:** Refer Q. 5.3.

**Q. 2. Explain how servlet works.**

**Ans:** Refer Q. 5.6.

**Q. 3. Explain cookies with its advantage and disadvantage.**

**Ans:** Refer Q. 5.9.

**Q. 4. What do you mean by JSP ? Explain the architecture of JSP.**

**Ans:** Refer Q. 5.12.

**Q. 5. Write a short note on :**

- a. Tomcat server
- b. Implicit objects in JSP
- c. Standard actions in JSP

**Ans:**

- a. Refer Q. 5.16.
- b. Refer Q. 5.19.
- c. Refer Q. 5.20.

**Q. 6. How error handling and debugging is done in JSP ?**

**Ans:** Refer Q. 5.15.

