Web Technology (KCS 602)

Course Outcome (CO)		**Bloom's Knowledge Level (KL)**
At the end of course , the student will be able to		
CO 1	Explain web development Strategies and Protocols governing Web.	K_1, K_2
CO 2	Develop Java programs for window/web-based applications.	K_2, K_3
CO 3	Design web pages using HTML, XML, CSS and JavaScript.	K_2, K_3
CO 4	Creation of client-server environment using socket programming	K_1, $K_{2,}$
CO 5	Building enterprise level applications and manipulate web databases using JDBC	$K_{3,}$ K_4
CO6	Design interactive web applications using Servlets and JSP	K_2, K_3

DETAILED SYLLABUS		**3-0-0**
Unit	**Topic**	**Proposed Lecture**
I	**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. Core Java: Introduction, Operator, Data type, Variable, Arrays, Methods & Classes, Inheritance, Package and Interface, Exception Handling, Multithread programming, I/O, Java Applet, String handling, Event handling, Introduction to AWT, AWT controls, Layout managers	**08**
II	**Web Page Designing:** HTML: List, Table, Images, Frames, forms, CSS, Document type definition, XML: DTD, XML schemes, Object Models, presenting and using XML, Using XML Processors: DOM and SAX, Dynamic HTML	**08**
III	**Scripting:** Java script: Introduction, documents, forms, statements, functions, objects; introduction to AJAX, **Networking :** Internet Addressing, InetAddress, Factory Methods, Instance Methods, TCP/IP Client Sockets, URL, URL Connection, TCP/IP Server Sockets, Datagram.	**08**
IV	**Enterprise Java Bean:** Preparing a Class to be a JavaBeans, Creating a JavaBeans, JavaBeans Properties, Types of beans, Stateful Session bean, Stateless Session bean, Entity bean **Java Database Connectivity (JDBC):** Merging Data from Multiple Tables: Joining, Manipulating, Databases with JDBC, Prepared Statements, Transaction Processing, Stored Procedures.	**08**
V	**Servlets:** Servlet Overview and Architecture, Interface Servlet and the Servlet Life Cycle, Handling HTTP get Requests, Handling HTTP post Requests, Redirecting Requests to Other Resources, Session Tracking, Cookies, Session Tracking with Http Session **Java Server Pages (JSP):** Introduction, Java Server Pages Overview, A First Java Server Page Example, Implicit Objects, Scripting, Standard Actions, Directives, Custom Tag Libraries..	**08**

Text books:

1. Burdman, Jessica, “Collaborative Web Development” Addison Wesley
2. Xavier, C, “ Web Technology and Design” , New Age International
3. Ivan Bayross,” HTML, DHTML, Java Script, Perl & CGI”, BPB Publication
4. Bhave, “Programming with Java”, Pearson Education
5. Herbert Schieldt, “The Complete Reference:Java”, McGraw Hill.
6. Hans Bergsten, “Java Server Pages”, SPD O’Reilly
7. Margaret Levine Young, “The Complete Reference Internet”, McGraw Hill.
8. Naughton, Schildt, “The Complete Reference JAVA2”, McGraw Hill.
9. Balagurusamy E, “Programming in JAVA”, McGraw Hill.

1 UNIT

Introduction and Core Java

CONTENTS

PART-1

Introduction : Introduction and Web Development Strategies, History of Web and Internet.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.1. **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, and computer platforms.
3. **Design for low bandwidth :** Web pages in website should be accessible at any connection speeds. If page downloaded 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. Reinforce the identifying elements of the site and create smooth transitions from one page to another.

Que 1.2. **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 which separates content structure from presentation.
5. In 1975, Alan Kay produces the first personal computer.

6. In 1987, CERN and the US laboratories connect to the internet as the main means of exchanging data between the laboratories.
7. In 1991 SLAC, the Stanford Linear Accelerator Center in California becomes the first web server in USA.
8. In 1992, the portable browser is released by CERN as freeware. Many HEP laboratories are now join with servers.
9. In 1995, Sun Microsystems produces HotJava, a browser which incorporates interactive objects.
10. In 2000, a massive denial of service attack is launched against major websites, including Yahoo, Amazon and e-bay.
11. In 2004, Abiline, the Internet2 backbone, upgraded from 2.5 Gbps to 10 Gbps. Network solutions begins offering 100 year domain registration.

Que 1.3. **Why it is important to identify the object in web development strategies ? Also explain, with the help of block diagram, web development process.**

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 :

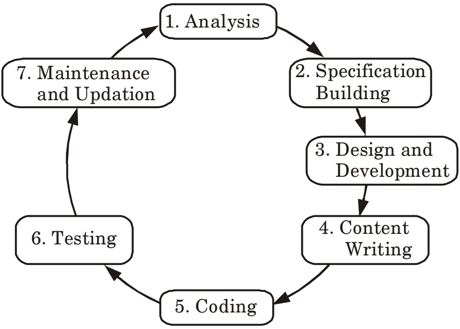

1. Analysis
2. Specification Building
3. Design and Development
4. Content Writing
5. Coding
6. Testing
7. Maintenance and Updation

Fig. 1.3.1. Block diagram.

1. **Analysis :**
   a. In this phase, all the information required for the web project is gathered.
   b. Analysis of the requirement given by the customer is done by the group of programmers and designers.

2. **Specification building :**

   a. In this phase, Software Requirement Specifications (SRS) document is prepared.

   b. In this document each and every element of the requirement is presented in detailed form.

3. **Design and development :**

   a. In design phase, the layouts and navigation will be designed as a prototype.

   b. Test plans and procedures are developed for quality assurance.

   c. In development phase, team will develop the database with all the data structures and sample data according to the requirements.

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 done in this phase.

5. **Coding :**

   a. In coding stage programmers add the code without disturbing the design.

   b. Coding team generate necessary testing plans as well as technical documentation.

6. **Testing :**

   a. Both automated testing and manual testing should be done without fail.

   b. After doing all the testing, a live testing is necessary for websites and web based applications. After uploading the site there should be a complete testing.

7. **Maintenance and updation :** Once the website is operational, ongoing promotion, technical maintenance, content management and staff training is needed on a regular basis.

PART-2

Protocol Governing Web, Writing Web Projects.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.4. **Describe in brief, the growth of the web technology. Explain the protocols governing the web.**

Answer

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

Protocols governing web :

1. **HTTP (Hypertext Transfer Protocol) :**
   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 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 (Internet Control Message Protocol) :**
   a. 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 (Routing Information Protocol) :**
   a. 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 (Open Shortest Path First) :**
   a. OSPF is classified as an Interior Gateway Protocol (IGP).
   b. 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.
5. **TCP/IP :**
   a. TCP/IP stands for Transmission Control Protocol / Internet Protocol.
   b. It 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. UDP :

a. User Datagram Protocol (UDP) is a connectionless protocol without any error detection facility.

b. It is also used for 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. Explain the HTTP protocol. Mention three basic features of HTTP that make HTTP a simple but powerful protocol. Give its architecture.

AKTU 2018-19, Marks 07

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. 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.

Architecture :

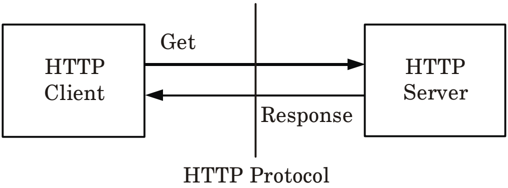


Fig. 1.5.1.

Que 1.6. Explain web project.

OR

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

Answer

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

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

Phase-I : Strategy :

1. Goals and objectives
2. Team building
3. Research and review
4. Project proposal

Phase-II : Design and specification :

1. Developing concepts
2. Content planning
3. Rough design
4. Final design
5. Build prototype
6. Prototype testing

Phase-III : Production or development :

1. Coding

Phase-IV : Testing and maintenance :

1. Automation testing
2. Manual testing

Phase-V : Register with ISP :

1. Buy domain name
2. Hoisting

Phase-VI : Launch

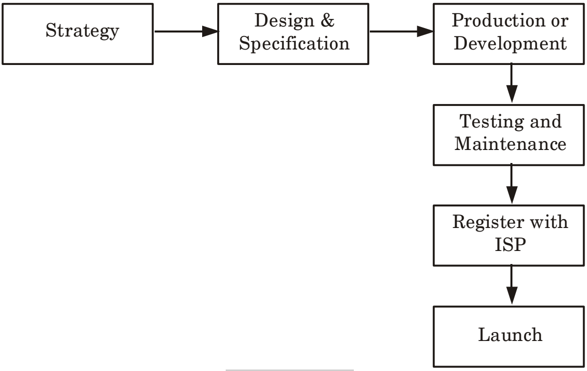


Fig. 1.6.1.

Que 1.7. What do you understand by web technologies ? Write down the steps to develop multi-department and large scale website.

OR

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

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 used in web design :

1. **Client-side technologies :**
   - a. **XHTML :**
     - i. XHTML is a markup language designed to structure information for presentation as web pages.
     - ii. All XHTML 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 XHTML 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/XSL :

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.

e. Linux/Apache : Linux is a popular open-source operating system, and Apache is the most widely used web server on the internet.

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

1. **Gathering information :** It includes purpose, main goals, and target audience.
2. **Planning :** It includes sitemap and wireframe creation.
3. **Design :** It includes page layouts and review,
4. Content writing and assembly
5. Coding
6. Testing, review, and launch
7. **Maintenance :** It includes monitoring and regular updating.

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 1.8. **Explain various protocols governing web. Also, explain web team.**

AKTU 2017-18, Marks 05

Answer

Protocols governing web : Refer Q. 1.4, Page 1–4D, Unit-1.

Web team :

1. Web team is a group of various technical experts in a developing site from coding the page to maintain the web server.
2. The ideal web team consists of two sub-teams :

   a. The client-side specialists, who create an attractive, clear front-end.

   b. The server-side specialists, who create a smoothly operating back-end.

Que 1.9. **Describe the objective of any website. Which type of essential skills required being a member of web project team ?**

AKTU 2018-19, Marks 07

Answer

Objectives of any websites :

1. Providing quality content on our website regularly, adding new information, establishing trust, marketing our site on other websites and social media to become an authoritative resource.
2. E-mail marketing lists, online support (live chat), webinars is provided to improve interaction with existing and potential customers.
3. Provide active social media program, promotions, reputation management to our regular customers to build our brand.

Essentials skills required being member of web project team are :

1. **HTML/CSS :** As a web developer, one should need to understand the basics of coding and markup language such as HTML and CSS.
2. **JavaScript :** After learning HTML and CSS, one should have the knowledge of JavaScript as it makes websites more interactive and functional.
3. **Photoshop :** For editing, designing, and stylizing websites one should have the knowledge of Photoshop to design a handful of banners and logos for clients.
4. **PHP language :** Other than HTML/CSS, one should have the skill of writing code in PHP language which is the core part of WordPress.

Que 1.10. **Discuss the basic elements of a good website design, including navigation considerations.**

Answer

Basic elements of a good website design are as follows :

1. **Original content :** The most important element is the content shown on the first page of website. The first page should clearly focus on the information about the business.
2. **Well-organized and easy to read :** The most important information on any page should be property organized and easy to read.
3. **More click, less scroll :** Website should be designed in such a way that it has more clickable element and less scrollable.
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 :** The website design should be easy to navigate and the menu items should easily accessible from any page. The viewer should always know exactly where they are on the website and have easy access to where they would like to be.
6. **Be interactive :** Create a site that encourages viewer participation.

Que 1.11. **Why planning is must before developing a website ? What are the advantages of early planning ?**

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 ?

PART-3

Connecting to Internet, Introduction to Internet Services and Tools, Introduction to Client-Server Computing.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.12. **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, 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 1.13. **What is internet service ? Explain various types of internet service.**

Answer

1. Internet service provides a way for data to be transferred from internet servers to our computer.
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
   b. Telnet
   c. Mailing lists
   d. Internet telephony (VoIP)
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.

4. **World Wide Web (WWW) :**

a. WWW offers a way to access documents spread over the several servers over the internet.

b. These documents may contain texts, graphics, audio, video, hyperlinks.

c. The hyperlinks allow the users to navigate between the documents.

Que 1.14. **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.

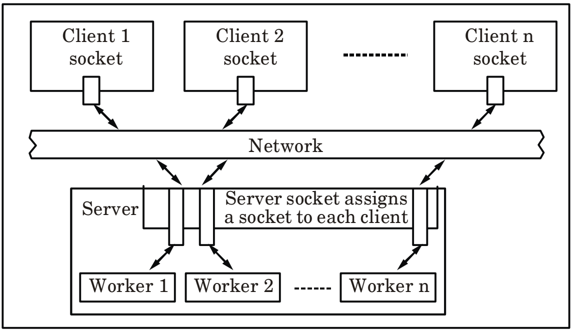


Fig. 1.14.1.

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-4

Core Java : Introduction, Operator, Data Type, Variable Arrays.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.15. **Explain some features of Java.**

OR

Elucidate the features of java. **AKTU 2016-17, Marks 15**

Answer

The basic features of Java are as follows :

1. **Simple :** Java is easy to learn.
2. **Object-oriented :** All the functions are defined inside the classes.
3. **Platform independent :** It is platform independent programming language because it compiles to byte-code. This byte code is interpreted by the Virtual Machine (JVM) on any platform.
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. So it is more secure.
6. **Portable :** Java is portable because it facilitates us to carry the Java bytecode to any platform. It does not require any implementation.
7. **Multithreaded :** Java programs are capable for easy to set up multiple threads of execution and coordinate parallel processes.
8. **Dynamic :** Java is a dynamic language. It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.
9. **Garbage collection :** Java has an automatic garbage collector which releases the objects which are not in use from long time.

Que 1.16. **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 1.17. **Discuss operators in Java.**

Answer

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

Following are the operators used in Java :

1. **Unary operator :**
   a. The Java unary operators require only one operand.
   b. Unary operators are used to perform various operations such as :
      i. Incrementing/decrementing a value by one (++,--)
      ii. Negating an expression (~)
      iii. Inverting the value of a Boolean (!)
2. **Arithmetic operator :** Java arithmetic operators are used to perform addition, subtraction, multiplication, and division. They act as basic mathematical operations.
3. **Shift operator :**
   a. **Left shift operator :** The left shift operator << is used to shift all of the bits in a value to the left side of a specified number of times.
   b. **Right shift operator :** The right shift operator >> is used to move left operands value to right by the number of bits specified by the right operand.

4. **Relational operator :** Relational operators enables us to test for any relationship between two operands. ==(equal to), != (not equal to), > (greater than), < (less than), >= (greater than or equal to), <= (less than or equal to) are the example of relational operator.

5. **Ternary operator :** Java ternary operator is used as one liner replacement for if-then-else statement and it is the only conditional operator which takes three operands.

6. **Assignment operator :** Java assignment operator is one of the most common operators. It is used to assign the value on its right to the operand on its left. =,+=,-=,*=, /=,%=, | = are the example of assignment operator.

7. **Bitwise operator :** Bitwise operator works on bits and performs bit-by-bit operation. It can be applied to the integer types, long, int, short, char, and byte. Binary AND operator, binary OR operator, binary XOR operator are the example of bitwise operator.

Que 1.18. **What is variable ? Describe different types of variable.**

Answer

1. A variable is a container which holds the value while the Java program is executed.
2. A variable is assigned with a data type.
3. Variable is a name of memory location.

There are three types of variables in java :

1. **Local variable :**
   a. A variable declared inside the body of the method is called local variable. We can use this variable only within that method and but not in the other methods.
   b. A local variable cannot be defined with "static" keyword.
2. **Instance variable :**
   a. A variable declared inside the class but outside the body of the method, is called instance variable.
   b. It is called instance variable because its value is instance specific and is not shared among instances.
3. **Static variable :**
   a. A variable which is declared as static is called static variable. It cannot be local.
   b. We can create a single copy of static variable and share among all the instances of the class.
   c. Memory allocation for static variable happens only once when the class is loaded in the memory.

Que 1.19. **Explain arrays in Java.**

Answer

1. Java array is an object which contains elements of a similar data type.
2. The elements of an array are stored in a contiguous memory location. It is a data structure where we store similar elements.
3. We can store only a fixed set of elements in a Java array.
4. The length of an array is established when the array is created.

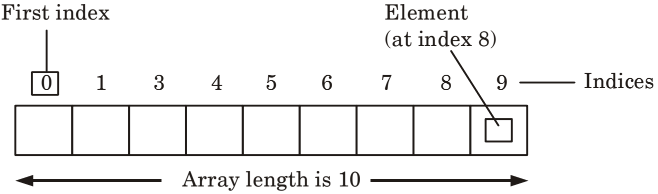


Fig. 1.19.1.

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

Creating, initializing, instantiation and accessing an array :

1. **Syntax to declare an array in Java :**

   dataType arrName[size];

2. **Syntax to initialize array in Java :**

   int a[ ] = {33, 3, 4, 5};

3. **Instantiation of an array in Java :**

   arrayRefVar = new datatype[size];

4. **Accessing an array element :** 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]);

PART-5

Methods and Classes, Inheritance, Packages and Interfaces.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.20. **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 1.21. **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 :

```
className    ObjectName;
```

Methods :

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

For example :

```
public class Test
{
int x = 5;
    public static void welcome ()
    {
        system.out.println (“ ------------------”);
    }
    public static void main(string args [ ])
    {
    welcome ();
    system.out.println(“Hi !!! This is my first program”);
    Test my obj = new Test ();
    System.out.println (“myobj. x”);
    }
}
```

Que 1.22. **Explain the concept of inheritance with its types.**

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.

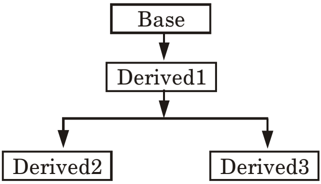


Fig. 1.22.1.

3. The inheritance relationship enables a derived class to inherit features from its base class. 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 :

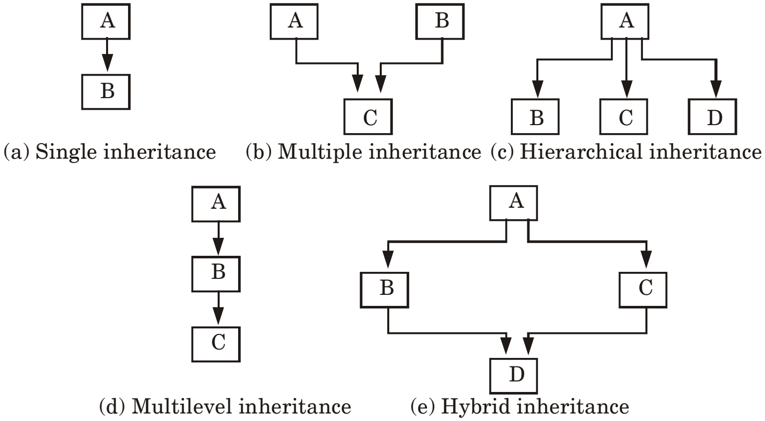
a. **Single inheritance :** It is the inheritance hierarchy wherein one derived class inherits from one base class.

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 wherein multiple subclasses inherit from one base class.

d. **Multilevel inheritance :** It is the inheritance hierarchy wherein subclass acts as a base class for other classes.

e. **Hybrid inheritance :** The inheritance hierarchy that reflects any legal combination of other four types of inheritance.



(a) Single inheritance (b) Multiple inheritance (c) Hierarchical inheritance

(d) Multilevel inheritance (e) Hybrid inheritance

Fig.1.22.2.

Que 1.23. **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. 1.21, Page 1–18D, Unit-1.

2. **Class :** Refer Q. 1.20, Page 1–18D, Unit-1.

3. **Inheritance :** Refer Q. 1.22, Page 1–19D, 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 access the data directly. 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 encapsulation shows that a non-member function cannot access an object's private or protected data.

Que 1.24. **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);
}
}
```

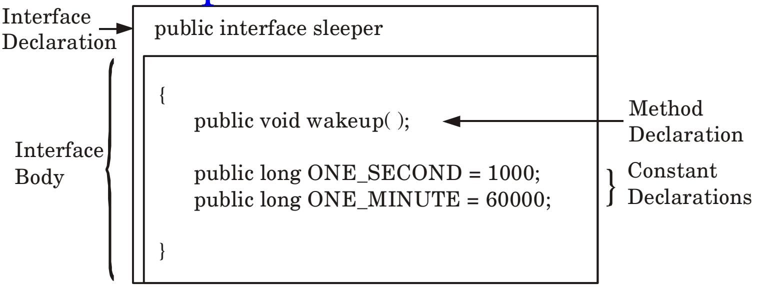
Que 1.25. **Write a short note on interface with example.**

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 1.25.1.



```
public interface sleeper
{
      public void wakeup( );

      public long ONE_SECOND = 1000;
      public long ONE_MINUTE = 60000;

}
```

Fig. 1.25.1.

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
```

Que 1.26. Compare object-oriented programming and object-based programming with example. List the features of object-oriented programming. Write a program in Java to demonstrate use of this keyword in constructor.

AKTU 2019-20, Marks 07

Answer

Comparison :

S. No.	Object-Oriented Programming	Object-Based Programming
1.	It supports all the features of OOPS.	It supports the usage of object and encapsulation.
2.	It also supports inheritance and polymorphism.	It does not support inheritance or, polymorphism or, both.
3.	It supports built-in objects.	It does not supports built-in objects.
4.	C#, Java, VB.Net are the examples of object-oriented languages.	JavaScript, VB are the examples of object-based languages.

Features of Object-Oriented programming are :

1. Inheritance
2. Polymorphism
3. Data hiding
4. Encapsulation
5. Overloading
6. Reusability

"this" keyword with constructor :

1. "this" keyword can be used inside the constructor to call another overloaded constructor in the same class. It is called the explicit constructor invocation.
2. This occurs if a class has two overloaded constructors, one without argument and another with the argument. Then "this" keyword can be used to call the constructor with an argument from the constructor without argument. This is required as the constructor cannot be called explicitly.

   Program :

```
class JBT {
JBT() {
this("JBT");
System.out.println("Inside Constructor without parameter");
}
JBT(String str) {
System.out.println("Inside Constructor with String parameter as " +
str);
}
public static void main(String[ ] args) {
JBT obj = new JBT();
}
}
```

Output :

```
Inside Constructor with String parameter as JBT
Inside Constructor without parameter
```

PART-6

Exception Handling, Multithread Programming, I/O.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.27. **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.

OR

What are exceptions and how they are handled ? Explain with an example. How we define a try and catch block ? Is it essential to catch all types of exceptions ? **AKTU 2018-19, Marks 07**

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.
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. 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.

    b. The catch block can have one or more statements that are necessary to process the exception.

    c. The catch statement is passed as a single parameter, which is reference to the type of exception object thrown by the try block.

    d. 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 :**

    a. 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.

    b. Finally block can be used to handle any exception generated within a try block.

    c. It may be added immediately after the try block or after the last catch block as follows :

```
    try                         try
    {                           {
         ..............              ..............
```

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

4. Throw :

a. Java supports "throw keyword" which is used if we want to throw our own exceptions.

b. We can do this by using the keyword throw as follows :

throw new Throwable_subclass;

No, it is not essential to catch all type of exceptions.

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.ArithmeticException: / by zero
finally block is always executed
rest of the code...
```

Que 1.28. **Define thread. How to create a thread in java ? Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 5.**

AKTU 2019-20, Marks 07

Answer

Thread : A thread is a light weighted process which runs concurrently with other threads. All threads of program define a separate path of execution.

Create a thread in Java :

```
public class Thread Test
{
public static void main (string [ ] args)
{
System.out.println ("Constructing the thread....");
BytePrinter bp = new BytePrinter ( );
System.out.println ("Starting the thread....");
bp start ( );
System.out.println ("The thread has been started");
System.out.println ("The main ( ) method is finishing");
return;
}
}
```

Program :

1. In the first step, we will implement the Runnable interface to define the logic of both threads. In the run method, we check if the number is even or odd.
2. If the number is even, we call the printEven method of the Printer class, else we call the printOdd method :

```
class TaskEvenOdd implements Runnable {
private int max;
private Printer print;
private boolean isEvenNumber;
// standard constructors
@Override
public void run() {
int number = isEvenNumber ? 2 : 1;
while (number <= max) {
if (isEvenNumber) {
print.printEven(number);
} else {
print.printOdd(number);
}
number += 2;
}
}
}
```

3. We define the Printer class as :

```
class Printer {
private volatile boolean isOdd;
synchronized void printEven(int number) {
while (!isOdd) {
try {
wait();
} catch (InterruptedException e) {
Thread.currentThread().interrupt();
}
}
System.out.println(Thread.currentThread().getName() + “:” +
number);
isOdd = false;
notify();
}
synchronized void printOdd(int number) {
while (isOdd) {
try {
wait();
} catch (InterruptedException e) {
Thread.currentThread().interrupt();
}
}
System.out.println(Thread.currentThread().getName() + ":" + number);
isOdd = true;
notify();
}
}
```

4. In the main method, we use the defined class to create two threads. We create an object of the Printer class and pass it as the parameter to the TaskEvenOdd constructor:

```
public static void main(String... args) {
Printer print = new Printer();
Thread t1 = new Thread(new TaskEvenOdd(print, 5, false),"Odd");
Thread t2 = new Thread(new TaskEvenOdd(print, 5, true),"Even");
t1.start();
t2.start();
}
```

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

Answer

1. A multithreaded program contains two or more parts that can run concurrently.
2. Each part of such a program is called a thread, and each thread defines a separate path of execution.
3. Each thread runs parallel to each other.
4. A multithreading is a specialized form of multitasking.
5. In multithread program, each thread has its own life cycle. The life cycle of thread is shown in Fig. 1.29.1.

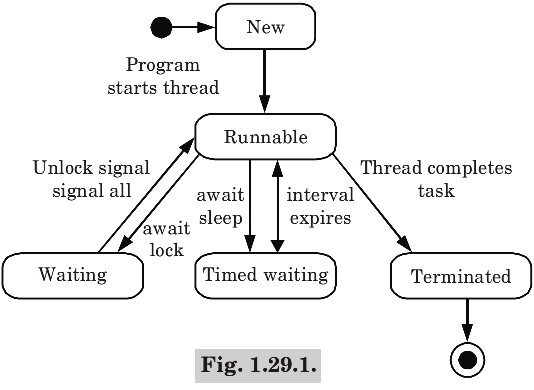


Fig. 1.29.1.

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 transition 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.

PART-7

Java Applet.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

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

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. Applets are used to make the website more dynamic and entertaining.
4. All applets are sub-classes (either directly or indirectly) of java.applet.Applet class.
5. 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.

d. **destroy( ) :** This method is only called when the browser shuts down normally.

e. **paint( ) :** Invoked immediately after the start( ) method, and also any time the applet needs to repaint itself in the browser.

Que 1.32. **Explain Applet with its life cycle. Write a program to demonstrate simple java applet to display any image. Compare Applets over HTML.**

AKTU 2019-20, Marks 07

Answer

Applet with its life cycle : Refer Q. 1.31, Page 1–32D, Unit-1.

Program :

```
import java.awt.*;
import java.applet.*;
public class DisplayImage extends Applet {
Image picture;
public void init() {
picture = getImage(getDocumentBase(),“sonoo.jpg”);
}
public void paint(Graphics g) {
g.drawImage(picture, 30,30, this);
}
}
```

Comparison :

S. No.	Applet	HTML
1	Applet is a JAVA program.	HTML is Hyper Text Markup Language.
2	It is embedded in the HTML code to provide functionality to the webpage.	It is used to provide position of the text in the webpage.
3.	It need Java package to run on JVM.	HTML do not need any package to run on browsers.

Que 1.33. **What is the difference between applet and application ? How is Java strongly associated with internet ? Draw a flowchart to show various Java tools are used in application development.**

AKTU 2018-19, Marks 07

Answer

Difference :

S. No.	Applets	Application
1.	An applet is a java program which is embedded in web page to generate dynamic content.	An application is a java program that runs independently on client/server without web browser.
2.	The execution of the program does not start from main( ) method.	The execution of the program starts from the main( ) method.
3.	Applet cannot run program from local machine.	Application can run the program from local machine.
4.	It is used to perform small tasks.	It is used to perform large tasks.
5.	It can only access browser specific services.	It can access all kind of services available on the system.

1. Java is strongly associated with the internet because the first application program written in Java was HotJava, a browser to run the applet on internet.
2. So, the internet users use the java to create the applet programs and run them locally using a java-enabled browser's like HotJava.
3. The users can use the java-enabled browsers to download the applet located on the computer system anywhere in the internet and run it on their computer.

Flowchart to show various Java tools used in application development :

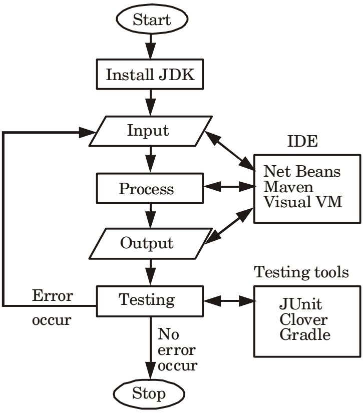


Fig. 1.33.1.

Que 1.34. **What are the advantages and drawback of applet ? Write a Java program to create an applet for calculator and also perform event handling on each button.** **AKTU 2018-19, Marks 07**

Answer

Advantages of applets :

1. Applets are platform independent.
2. Applets are quite secure and safe to use.
3. Applets cache quickly.
4. Applet increase interactivity for users.
5. Database integration is another important advantage of applets.

Drawbacks of applets :

1. Applets do not access client-side resources, like such as file, operating system.
2. Applet cannot work with native methods.
3. Applet can only extract information about client-machine *i.e.*, its name, Java version, OS, version etc.
4. Mobile browsers which are running on IOS or Android do not support applets.

Program :

```
import java.applet.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.TextEvent;
import java.awt.event.TextListener;;
public class calculator extends Applet implements ActionListener,
TextListener {
String s,s1,s2,s3,s4;
Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0;
Button add,sub,eq,cl,mul,div;
TextField t1;
int a,b,c;
public void init() {
```

```
t1=new TextField(10);
b1=new Button(“1”);  b2=new Button(“2”); b3=new Button(“3”);
b4=new Button(“4”); b5=new Button(“5”); b6=new Button(“6”);
b7=new Button(“7”);  b8=new Button(“8”); b9=new Button(“9”);
b0=new Button(“0”);
add=new Button(“+”); sub=new Button(“–”);
mul=new Button(“*”); div=new Button(“/”);
eq=new Button(“=”); cl=new Button(“Clear”);
GridLayout gb=new GridLayout(4,5);
setLayout(gb);
add(t1);add(b1);add(b2); add(b3);add(b4);add(b5);
add(b6); add(b7);add(b8); add(b9);add(b0);add(add);
add(sub);add(mul);add(div); add(eq);add(cl);
b1.addActionListener(this); b2.addActionListener(this);
b3.addActionListener(this); b4.addActionListener(this);
b5.addActionListener(this); b6.addActionListener(this);
b7.addActionListener(this); b8.addActionListener(this);
b9.addActionListener(this); b0.addActionListener(this);
add.addActionListener(this); sub.addActionListener(this);
mul.addActionListener(this); div.addActionListener(this);
eq.addActionListener(this); cl.addActionListener(this);
paint();
//t1.addTextListener(this); }
public void paint() {
setBackground(Color.green); }
public void actionPerformed(ActionEvent e) {
s=e.getActionCommand();
if(s.equals(“0”) || s.equals(“1”) || s.equals(“2”) ||
s.equals(“3”) || s.equals(“4”) || s.equals(“5”) || s.equals(“6”) ||
s.equals(“7”) || s.equals(“8”) ||
s.equals(“9”) || s.equals(“0”)) {
s1=t1.getText()+s;
t1.setText(s1); }
```

```
if(s.equals(“+”)) {
s2=t1.getText();
t1.setText(“”);
s3=“+”; }
if(s.equals(“–”)) {
s2=t1.getText();
t1.setText(“”);
s3=“–”; }
if(s.equals(“*”)) {
s2=t1.getText();
t1.setText(“”);
s3=“*”; }
if(s.equals(“–”)) {
s2=t1.getText();
t1.setText(“”);
s3=“–”; }
if(s.equals(“=”)) {
s4=t1.getText();
a=Integer.parseInt(s2);
b=Integer.parseInt(s4);
if(s3.equals(“+”))
c=a+b;
if(s3.equals(“–”))
c=a–b;
t1.setText(String.valueOf(c)); }
if(s.equals(“Clear”)) {
t1.setText(“”); } }
public void textValueChanged(TextEvent e) { }}
```

PART-8

String Handling, Event Handling.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.35. **Explain string handling in Java with example.**

Answer

1. String handling is a process of performing different operation such as concatenation, comparison on the string.
2. Following are the method used in string handling :
   - a. **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.
   - b. **Java string compareTo() :** The Java string compareTo() method compares the given string with current string. It returns positive number, negative number or zero.
   - c. **Java string concat() :** The Java string concat() method combines a specific string at the end of another string and returns a combined string.
   - d. **Java string replace() :** The Java string replace() method returns a string, replacing all the old characters to new characters.
   - e. **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.
   - 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 :

```
public class Example{
public static void main(String args[]{
String s1=“Hello”;
String s2=“Aditya”;
s3=s1.concat(“how are you?”);
System.out.println(“Length of string s1 is: ”+s1.length());
System.out.println(s1.compareTo(s2));
System.out.println(s3);
System.out.println(s3.contains(“you”));
System.out.println(s1.replace(‘H’, “T”));
}}
```

Output :

```
Length of string s1 is: 5
False
Hello how are you?
True
Tello
```

Que 1.36. Explain event handling in brief.

Answer

Event :

1. An event happens when something changes within a graphical user interface.
2. We can say that events are objects in Java. It comes under some classes stored in java.util.EvenObject.
3. The Abstract Window Toolkit (AWT) uses event driven programming to do processing of user actions, one that underlies all modern window systems programming.
4. An event describes as a particular user action.
5. The Java run time notifies the program when an interesting event occurs.
6. For example, events occur when a user clicks on a button, clicks on a combobox, or types characters into a text field, such as in the following:
   a. For a button, the event that is fired is the ActionListener.
   b. For a text field, it is the KeyEvent.

The following is required to perform event handling :

1. Implement the Listener interface and override its methods
2. Register the component with the listener

Following are the public methods used for various components :

1. **Button :** void addActionListener(ActionListener a)
2. **List :**

   void addActionListener(ActionListener a) void addItemListener(ItemListener a)

3. **Choice :**

   void addItemListener(ItemListener x)

4. **MenuItem :**

   void addActionListener(ActionListener x)

5. **TextField :**

   void addActiontListener(ActionListener x)

   void addTextListener(TextListener x)

6. **TextArea :**

   void addTextListener(TextListener x)

PART-9

Introduction to AWT, AWT Controls, Layout Managers.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 1.37. Write short note on AWT.

AKTU 2016-17, Marks 05

Answer

1. The AWT stands for Abstract Window Toolkit.
2. AWT is a library of class which provides GUI tools to develop GUI application and applet.
3. It provides many classes for programmers to use. It is the connection between our application and the native GUI.
4. 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 :
   a. **Containers class :** Frame, Dialog, Panel, Applet etc.
   b. **Components class :** TextField, Button, Checkbox, Scrollbar, Label, List etc.
   c. **Custom graphics class :** Colour, Font, Dimensions etc.

6. The AWT supports following types of controls :
   - i. Buttons
   - ii. Checkbox
   - iii. CheckboxGroup
   - iv. Choice
   - v. Label
   - vi. List
   - vii. Scrollbar
   - viii. TextField
   - ix. TextArea

Que 1.38. **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. 1.37, Page 1–40D, Unit-1.

1. **Canvas :** A canvas is a graphical component representing a region where we can draw things such as rectangles, circles, and text strings.
2. **Checkbox :**
   - a. A Checkbox is a label with a small push button.
   - b. The state of Checkbox is either true (button is checked) or false (button not checked).
   - c. The default initial state is false.
3. **CheckboxGroup :**
   - a. A CheckboxGroup is used to control the behaviour of a group of Checkbox objects (each of which has a true or false state).
   - b. Exactly one of the Checkbox objects is allowed to be true at one time.
   - c. Checkbox objects controlled with a CheckboxGroup are usually referred to as "radio buttons".
4. **TextArea :**
   - a. A TextArea is a multi-row text field that displays a single string of characters, where newline ('\n' or 'n\n\r' or '\r', 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 1.39. **Explain LayoutManager in brief.**

OR

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

Answer

LayoutManager :

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

Types of LayoutManager :

a. **BorderLayout :** This scheme lays out the component in five ways :
   i. North-Northern part of the container
   ii. South-Southern part of the container
   iii. East-Eastern part of the container
   iv. West-Western part of the container
   v. Center-centered in the container

b. **CardLayout :** Allows for what Windows programmers have called "tabbed dialogs" or dynamic dialogs.

c. **GridLayout :** Allows for the layout of components in a grid-like fashion rather than "North" or "Center".

d. **FlowLayout :** Allows for component to be laid out in a row (or flow) and aligned (left, right, center).

e. **None :** No layout, the container will not attempt to reposition the components during an update.

Que 1.40. **Explain AWT and its controls. How the layout manager manage the AWT controls ? Write a program to demonstrate graphics (*i.e.* line, circle, rectangle etc.) using Frame, Panel, and layout manager.**

AKTU 2019-20, Marks 07

Answer

AWT : Refer Q. 1.37, Page 1–40D, Unit-1.

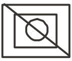
AWT controls : Refer Q. 1.38, Page 1–41D, Unit-1.

Layout manager : Refer Q. 1.39, Page 1–42D, Unit-1.

Program :

```
package com.mkyong;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.Shape;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Line2D;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
public class DrawShapes extends JFrame {
private static final long serialVersionUID = 1L;
public DrawShapes() {
setSize(new Dimension(320, 320));
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
JPanel p = new JPanel() {
@Override
public void paintComponent(Graphics g) {
Graphics2D g2 = (Graphics2D) g;
Shape line = new Line2D.Double(3, 3, 303, 303);
Shape rect = new Rectangle(3, 3, 303, 303);
Shape circle = new Ellipse2D.Double(100, 100, 100, 100);
Shape roundRect = new RoundRectangle2D.Double(20, 20, 250, 250,
5, 25);
g2.draw(line);
g2.draw(rect);
g2.draw(circle);
g2.draw(roundRect);
}
};
setTitle("My Shapes");
this.getContentPane().add(p);
}
public static void main(String arg[ ]) {
SwingUtilities.invokeLater(new Runnable() {
@Override
public void run() {
// TODO Auto-generated method stub
new  DrawShapes();
}
});
}
}
```

Output :



Que 1.41. **What are the uses of layout managers ? Give the name of classes that represents the layout managers. Explain any five layout managers.** **AKTU 2018-19, Marks 07**

Answer

Layout manager is used to place the component such as buttons, text boxes on the application.

Classes that represent the layout managers are :

1. BorderLayout
2. FlowLayout
3. GridLayout
4. CardLayout
5. GridBagLayout
6. BoxLayout
7. NoLayout
8. ScrollPaneLayout
9. SpringLayout

Five layout managers : Refer Q. 1.39, Page 1–42D, Unit-1.

VERY IMPORTANT QUESTIONS

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

Q. 1. Why it is important to identify the object in web development strategies ? Also explain, with the help of block diagram, web development process.

Ans. Refer Q. 1.3.

Q. 2. Explain the HTTP protocol. Mention three basic features of HTTP that make HTTP a simple but powerful protocol. Give its architecture.

Ans. Refer Q. 1.5.

Q. 3. Describe the objective of any website. Which type of essential skills required being a member of web project team ?

Ans. Refer Q. 1.9.

Q. 4. Explain some features of Java.

Ans. Refer Q. 1.15.

Q. 5. Discuss the key features supported by object-oriented programming languages.

Ans. Refer Q. 1.23.

Q. 6. What are packages in java ? How a user-defined package is created in java, explain with example ?

Ans. Refer Q. 1.24.

Q. 7. Compare object-oriented programming and object-based programming with example. List the features of object-oriented programming. Write a program in Java to demonstrate use of this keyword in constructor.

Ans. Refer Q. 1.26.

Q. 8. What are exceptions and how they are handled in java ? Explain the keywords try, catch, throw, and finally with example.

Ans. Refer Q. 1.27.

Q. 9. Explain Applet with its life cycle. Write a program to demonstrate simple java applet to display any image. Compare Applets over HTML.

Ans. Refer Q. 1.32.

Q. 10. What are the advantages and drawback of applet ? Write a Java program to create an applet for calculator and also perform event handling on each button.

Ans. Refer Q. 1.34.

Q. 11. Explain AWT and its controls. How the layout manager manage the AWT controls ? Write a program to demonstrate graphics (*i.e.* line, circle, rectangle etc.) using Frame, Panel, and layout manager.

Ans. Refer Q. 1.40.

☺☺☺

2

UNIT

Web Page Designing

CONTENTS

PART-1

Web Page Designing : HTML : List, Tables, Images.

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

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 using HEAD tags :**

      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 in the head section are :

         a. Title tag

         b. Style tag

   b. **Document body using BODY tags :**

      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 : BGCOLOUR, 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.

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 are shows up in the title bar of the browser window.
   For example :
   <TITLE> Quantum Page </TITLE>
2. **Footer :**
   a. Certain information is needed to be placed at the foot of the HTML document like copyright information, contact details etc. This can be achieved by using <ADDRESS> - - - - </ADDRESS> tags.
   b. This tag should ideally be placed immediately after the last line of the textual material of the web page.
   For example :
   <ADDRESS> Copyright © QUANTUM PAGE PVT. LTD. </ADDRESS>
3. **Paragraph breaks :**
   a. <P> tag is used to break paragraph or skip one line between the 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.

c. Browsers recognize multiple consecutive <BR> tags.

Example : <P> For further details contact us at <BR> QUANTUM PAGE PVT. LTD.<BR>GHAZIABAD<BR>UTTAR PRADESH</ P>.

Output :

```
For further details contact us at
QUANTUM PAGE PVT. LTD.
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 example.**

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>.

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 : This attribute specifies the alignment of the horizontal rule. It can have three values as left, right and center.
      ii. SIZE : Changes the height of the horizontal rule.
      iii. WIDTH : Sets the width of the horizontal rule.

3. **Text styles :** For text styling we use three different tags :
   a. Bold : Displays text in boldface style. The tags used are <B> - - - - </B>.
   b. Italic : Displays text in italic. The tags used are <I> - - - - </I>.
   c. Underline : Displays text as underlined. The tags used are <U> - - - - </U>.

For example :

```
<BODY>
<B> Welcome to Quantum Page. </B>
<I> Welcome to Quantum Page. </I>
<U> Welcome to Quantum Page. </U>
</BODY>
```

Output :

Welcome to Quantum Page.

Welcome to Quantum Page.

<u>Welcome to Quantum Page.</u>

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

Example : <CENTER> Welcome to Quantum Page! </CENTER>

Output : Welcome to Quantum Page!

5. Font setting tags :

a. All text specified within the tags <FONT> and </FONT> will appear in the browser. The style of the text depends on the attributes of the <FONT>.

b. The attributes are :

i. FACE : Sets the font to the specified font family.

ii. SIZE : Sets the size of the text. SIZE can take values between 1 and 7. The default size used is 3.

iii. COLOUR : Sets the colour of the text. COLOUR can be written in an English language colour name or in hexadecimal triplet form.

Example : <FONT FACE = “Arial” SIZE = “6” COLOUR = “blue”>

Welcome to Quantum Publication. </FONT>

Que 2.4. What is list in HTML ? What are the different types of lists in HTML ? Give an example of each type.

Answer

Lists are used to present the list of information in well formed and semantic way.

Types of list are :

1. Unordered list :

i. An unordered list starts with the tag <UL> and ends with </UL>.

ii. Each list item starts with the tag <LI>.

iii. 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.

iv. 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. A definition list is a list of terms and corresponding definitions.

ii. Definition list values appear within <DL> and </DL> tags.

iii. Definition lists consist 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. The title of a table in the HTML document is done using table caption.
   - 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. **BORDER :** Controls the border to be placed around the table. The border thickness is specified in pixels.
   - c. **CELL PADDING :** Controls the distance between the data in a cell and the boundaries of the cell.
   - d. **CELL SPACING :** Controls the spacing between adjacent cells.
   - e. **COLSPAN :** Width of the cell in terms of number of columns is used when a cell occupies more than one column.
   - f. **ROWSPAN :** Height of the cell in terms of rows is 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.**

OR

Explain image is HTML.

Answer

1. HTML accepts many picture file formats such as .png, .gif and .jpg (.jpeg) etc.
2. To add an image to web page we used the <IMG> tag, which takes the name of the image file as an attribute, also control the height, width, border etc.
3. The <IMG> tag takes the following attributes :
   - **a. ALT :** Indicates the text to be displayed in case the browser is unable to display the image specified in the SRC attribute.
   - **b. SRC :** Specifies the location and name of the image file.
   - **c. WIDTH :** Specifies the width of the image in pixels.
   - **d. HEIGHT :** Specifies the height of the image in pixels.
   - **e. 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.
   - **f. BORDER :** Specifies the size of the border to place around the image.

Creating an image link :

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

Example : <A HREF = "QUANTUM.HTML">

<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 *i.e.*, QUANTUM.HTML.

Que 2.7. **Create an html page named as "Table.html" to display your class time table.**

i. Provide the title as Time Table.

ii. Provide various colour options to the cells (Highlight the lab hours and elective hours with different colours).

AKTU 2018-19, Marks 07

Answer

hyperlink.html :

```
<html><body>
```

```
<a href=“Pagelink.html”>Different Tags</a><br>
<br><a href=“table.html”>Time Table</a><br>
</body></html>
```

Table.html :

```
<html><head>
<title>Time Table</title>
</head><body>
<B><center><h3>
AKTU UNIVERSITY,<br>
DEPARTMENT OF INFORMATION TECHNOLOGY<br>
EFFECTIVE FROM : 04.02.19<br></center></b></h3>
<table border=“2” width=“75%” cellspacing=“2” cellpadding=“25”
align=center><h3>
<caption align=left>Sem: III IT\CS
& nbsp; & nbsp; & nbsp; & nbsp; & nbsp; &nbsp; & nbsp; & nbsp; & nbsp;
& nbsp; & nbsp;  & nbsp; & nbsp; & nbsp; & nbsp; & nbsp; & nbsp; & nbsp;
& nbsp; & nbsp; & nbsp; & nbsp; </caption></h3>
<thead><tr>
<th>Periods</th>
<th ROWSPAN=2>1 <br>8.30-9.20</th>
<th ROWSPAN=2>2<br>9.20-10.10</th>
<th ROWSPAN=2>3<br>10.20-11.10</th>
<th ROWSPAN=2>4<br>11.10-12.00</th>
<th ROWSPAN=2>5<br>1.30-2.20</th>
<th ROWSPAN=2>6<br>2.20-3.10</th>
<th ROWSPAN=2>7<br>3.10-4.00</th>
</tr>
<tr><th> Days</th>
</thead>
<tbody><tr>
<th>Monday</th>
<td>IT0302</td><td>IT0306</td><td>IT0304</td>
IT0322 -Web Technology LAB
<td>IT0306</td><td>IT0302</td>
<td COLSPAN=2><center>IT0324</td></tr>
<tr>
<th>Tuesday</th>
<td COLSPAN =4 bgcolor=“cyan”><center>IT0320/IT0322</td>
<td>IT0310</td>
<td>IT0304</td><td>IT308</td></tr>
<tr>
<th>Wednesday</th>
<td COLSPAN =2 bgcolor=“red”><center>ELECTIVE - I</td>
<td>IT0306</td><td>IT310</td>
<td>IT304</td><td>IT308</td>
<td>COUN</td></tr>
<tr>
```

```
<th>Thursday</th>
<td>IT302</td><td>IT0304</td>
<td COLSPAN=2><center>PD0302</td>
<td COLSPAN=3 bgcolor="cyan"><center>IT0320/IT0322</td>
</tr>
<tr>
<th>Friday</th>
<td>IT0308</td><td>IT0306</td>
<td>IT0308</td><td>IT0302</td>
<td COLSPAN=2 bgcolor="red"><center>ELECTIVE - I</td>
<td></td>
</tr></tbody></table></body>
```

PART-2

Frames.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.8. **Explain frames in HTML with example.**

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.

Following two tags are used in HTML for frame :

a. <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.

b. <FRAME> tag :

1. Once the browser screen is divided into rows and columns, each unique section defined can be loaded with different HTML documents.
2. This is achieved by using the <FRAME> tag, which takes the following attributes :

a. SRC = "url" : Indicates the URL of the document to be loaded into the frame.

b. MarginHeight = "n" : Specifies the amount of white space to be left at the top and bottom of the frame.

c. MarginWidth = "n" : Specifies the amount of white space to be left along the sides of the frame.

d. Name = "name" : Gives the frame a unique name so it can be targeted by other documents. The name given must begin with an alphanumeric character.

e. Noresize : Disables the frames resizing capability.

f. Scrolling : Controls the appearance of horizontal and vertical scrollbars in a frame. This takes the values YES/NO/AUTO.

For 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 2nd frame with File4.html.

</FRAMESET>

</FRAMESET>

</HTML>

Que 2.9. **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.**

Answer

```
<html>
<FRAME ROWS = “50%, 50%”>
<FRAMESET COLS =“50%, 50%”>
<FRAME SRC=“File 1.html”> </FRAME>
<FRAME SRC=“File 2.html”> </FRAME>
</FRAMESET>
<FRAME SRC =“File 3.html”> </FRAME>
</FRAMESET>
</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>
```

Que 2.10. **Discuss how frames play a big role in advertising on web. What roles do form play in making web page dynamic ?**

AKTU 2018-19, Marks 07

Answer

Role of frames in advertising on web :

1. A frame is a part of a web page or browser window which displays content independent of its container, with the ability to load content independently.
2. Frames play a big role in advertising on web by allowing the following advantages :
   a. Content can be loaded and navigated independently.
   b. Simple maintenance of content shared across all or most pages.
   c. Reducing the amount of bandwidth needed.
   d. Allowing several pieces of information to be viewed side by side.
3. All the above advantages helps in reducing the advertising budget and the audience can view the intended information easily.

Roles of form in making web page dynamic :

1. In hidden frame technique we could hide or minimise the size of frame.
2. The hidden frame is loaded with a web page that contains a form, and JavaScript is used to dynamically fill out the form making the web page dynamic.

Que 2.11. **Using a frameset, create an HTML document like following.**

<table>
<tr><th colspan="2">Header.html</th></tr>
<tr><td>Menu.html</td><td>Output.html</td></tr>
</table>

Also host it as a web page on any server.

Answer

Frameset.html :

```
<!DOCTYPE html>
<html>
<frameset rows="25%,*">
 <frame src="Header.htm">
<frameset cols="50%,50%">
<frame src="Menu.htm">
<frame src="Output.htm">
</frameset>
</frameset>
</html>
```

Node.js :

```
var http = require('http');
var url = require('url');
var fs = require('fs');
var server = http.createServer(function(request, response) {
 var path = url.parse(request.url).pathname;
switch (path) {
case '/':
response.writeHead(200, {
'Content-Type': 'text/plain'
});
response.write("This is Test Message.");
response.end();
break;
case '/frameset.html':
fs.readFile(__dirname + path, function(error, data) {
if (error) {
response.writeHead(404);
response.write(error);
response.end();
} else {
response.writeHead(200, {
'Content-Type': 'text/html'
});
response.write(data);
response.end();
}
});
break;
default:
response.writeHead(404);
response.write(“opps this does not exist - 404”);
response.end();
break;
}
});
server.listen(8082);
```

PART-3

Forms.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.12. 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 :** This attribute is used to specify where the form data is to be sent to the server after submission of the form.
   - **b. METHOD :**
     - i. This attribute specifies how to send form data.
     - ii. The form data can be sent as URL variables (with method = “GET”) or as HTTP post transaction (with method = “POST”).

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. 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="Submit">

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.

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. Dropdown list :

a. <SELECT> tag is used to create a dropdown list.

b. 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. Reset the form :

a. The input type RESET 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.

Que 2.13. **Design a HTML form for a railway reservation system.**

AKTU 2016-17, Marks 10

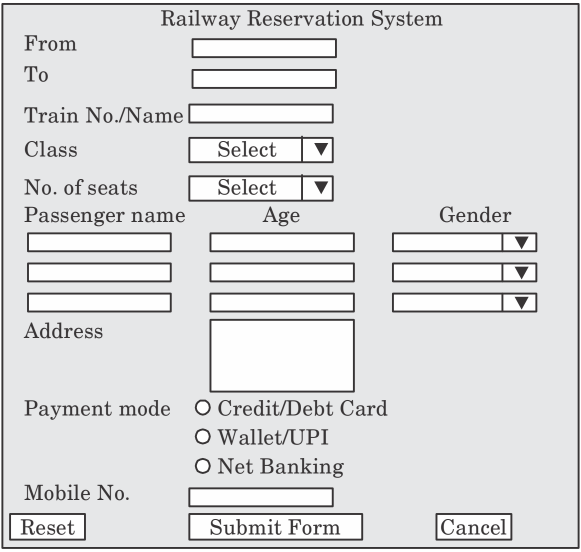
Answer

```
<html>
<head>
<script type = “text/JavaScript” src = “validate.js”></script>
</head>
<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>
<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>
</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>
<tr><td>Age</td></tr>
<tr><td>Gender</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 = “P2”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”>Wallet
UPI</td>
<td><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>
<td>
<input type=“Cancel” value=“Cancel”>
</td>
</tr>
</table>
</form>
</body>
</html>
```

Output :



Railway Reservation System

From

To

Train No./Name

Class Select ▼

No. of seats Select ▼

Passenger name Age Gender

Address

Payment mode ○ Credit/Debt Card

○ Wallet/UPI

○ Net Banking

Mobile No.

Reset Submit Form Cancel

PART-4

CSS.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.14. 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.

Que 2.15. Explain the features of CSS.

Answer

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

AKTU 2017-18, Marks 10

Answer

CSS : Refer Q. 2.14, Page 2–21D, 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>
```

Que 2.17. **What do you mean by CSS ? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system. Mention how can you integrate CSS on a web page ?**

AKTU 2018-19, Marks 07

Answer

CSS and its integration on a web page : Refer Q. 2.16, Page 2–22D, Unit-2.

Program :

```
<HTML><HEAD><STYLE>
H1 {colour: red; font-family: arial; font-size: 2.50 em}
</STYLE></HEAD><BODY>
<H1> This is the H1 element </H1>
</BODY></HTML>
```

Que 2.18. **Describe the role and importance of CSS in web designing. Also differentiate Class and Id in CSS.**

Answer

Role of CSS :

1. It allows us to develop the overall look of our website.
2. It allows us to position and reposition the components of a web page with relative ease.
3. It allows us to alter and control the every aspect of layout in a web page.

Importance of CSS :

1. It helps in quick loading of web pages.
2. It helps in saving bandwidth.
2. It helps in customization of web pages.
3. It provides complete consistency to the web pages.

Difference :

S. No.	Class selector	Id selector
1.	Class can be used to identify more than one element.	Id can be used to identify one element.
2.	In CSS a class selector is a name preceded by a full stop (".")	In CSS an Id selector is a name preceded by a hash character ("#").
3.	**Syntax :** .class_selector_name {property : value;}	**Syntax :** #id.selector–name {property : value;}

Que 2.19. **Explain CSS. What are the CSS frameworks? Explain in brief. What are the different ways of using the stylesheet? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system.** **AKTU 2019-20, Marks 07**

Answer

CSS : Refer Q. 2.14, Page 2–21D, Unit-2.

CSS frameworks :

A CSS framework is a library allowing web design for easier, more standards-compliant using the Cascading Style Sheets language. Bootstrap, Foundation, Bulma, UIkit, Semantic UI are some of the CSS frameworks.

Different ways of using stylesheet : Refer Q. 2.16, Page 2–22D, Unit-2.

Program : Refer Q. 2.17, Page 2–23D, Unit-2.

Que 2.20. **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, italic 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 text and 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 possible values are capitalize uppercase, lowercase etc.
   e. **Text-decorate :** Denotes the text decoration. The standard values 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-rigid, margin-bottom, or 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.
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-5

Document Type Definition, XML : DTD.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.21. Discuss various types of DTDs (Document Type Definition) in XML. Which type of DTD is preferable and why ?

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 root-element [element-declarations]>
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 file.

Que 2.22. What is DTD ? What are the differences between external and internal DTD ? Use suitable example.

AKTU 2015-16, Marks 10

Answer

DTD : Refer Q. 2.21, Page 2–26D, Unit-2.

Difference :

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 : <!DOCTYPE root-element SYSTEM “file-name”> where file-name is the file with .dtd extension.	The syntax for internal DTD is : <!DOCTYPE root-element [element-declarations]> 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)>
<!ELEMENT phone (#PCDATA)> ]>
<address>
<name>Pratibha </name>
<company>Quantum</company>
<phone>(011)  123-4567</phone>
</address>
```

Example of external DTD :

```
<?xml version=“1.0” encoding="UTF-8" standalone=“no” ?>
<!DOCTYPE address SYSTEM “address.dtd”>
<address>
```

```
<name>Prabha 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 2.23. **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. 2.21, Page 2–26D, Unit-2.

Que 2.24. **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

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>
<price>29.99</price>
</book>
<book category=“web” cover=“paperback”>
<title lang=“en”>Learning XML</title>
<author>Erik T. Ray</author>
<year>2013</year>
<price>39.95</price>
</book>
</bookstore>
```

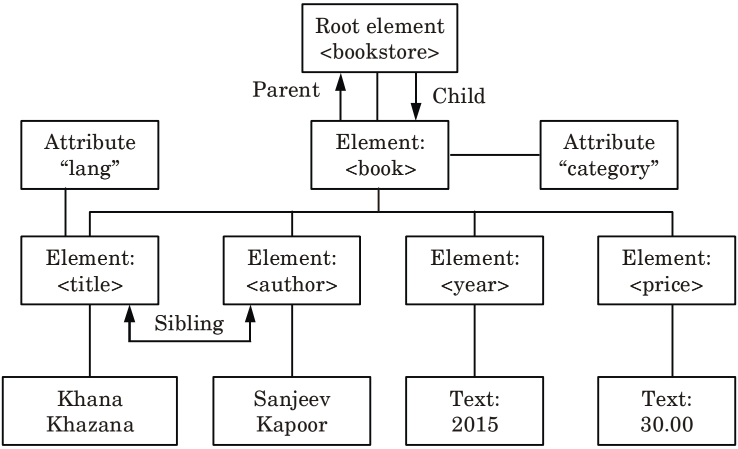


Fig. 2.24.1.

XML parser is useful in :

1. Loading the elements of XML document.
2. Accessing the elements of XML document.
3. Deleting the elements of XML document.
4. Changing the elements of XML document.

Que 2.25. What is XML ? Create a XML document of 10 students of final CSE. Add their roll number, marks obtained in 5 subjects, total marks and percentage. Save this XML document at the server, write a program that accepts student's roll number as input and returns the students marks, total percentage by taking student information for XML document.

AKTU 2018-19, Marks 07

OR

Discuss DTD. How the DTD is different from XSD ? Demonstrate to create a XML document of 10 students of third year. Add their roll numbers, marks obtained in 5 subjects, total marks and percentage and validate using DTD.

AKTU 2019-20, Marks 07

OR

Discuss XML. Which technology is used to define the structure of XML document ? Explain and demonstrate with an example.

AKTU 2019-20, Marks 07

Answer

XML : Refer Q. 2.24, Page 2–28D, Unit-2.

DTD : Refer Q. 2.21, Page 2–26D, Unit-2.

Program :

XML document :

Student.XML

```
<?xml version = "1.0"?>
<!DOCTYPE STUDENTS SYSTEM "E:\XML1\STUDENT.dtd">
<STUDENTS><STUDENT><STUDENTDATA>
<Roll_No> 001</Roll_No>
<NAME> RAM</NAME>
<Marks>
<Marks 1> 70 </Marks 1>
<Marks 2> 80 </Marks 2>
<Marks 3> 50 </Marks 3>
<Marks 4> 60 </Marks 4>
<Marks 5> 70 </Marks 5>
<Total> 330 </Total>
<Percentage> 66.0 </Percentage>
</Marks>
</STUDENTDATA></STUDENT></STUDENTS>
```

> **NOTE :** *Write the above code (bold part only) nine times again to enter the details of rest of the nine students.*

Student.dtd

```
<?xml version "1.0"?>
<!ELEMENT  STUDENTS (STUDENT*)>
<!ELEMENT  STUDENT (STUDENTDATA*)>
<!ELEMENT  STUDENTDATA (Roll_No, Name, Marks, Total, Percentage)>
<!ELEMENT  Roll_No (#PCDATA)>
<!ELEMENT  NAME (#PCDATA)>
<!ELEMENT  Marks (#PCDATA)>
<!ELEMENT  Total (#PCDATA)>
<!ELEMENT  Percentage (#PCDATA)>
```

Student.html

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"E:\XML 1\STUDENT.dtd"">

<html>
```

```
<head COLOUR:RED><h1 style = “color : red”>
<MARQUEE DIRECTION = “RIGHT”><CENTER> COLLEGE OF
ENGINEERING AND TECHNOLOGY </CENTER> </MARQUEE> </H1>
<title> STUDENT DETAILS DISPLAY <title>
</head>
<body style = “background-colour : PINK”> <H2 STYLE = “COLOUR :
BLUE”> <MARQUEE> <CENTER> DEPARTMENT OF COMPUTER
SCIENCE
</CENTER> </MARQUEE><BR></H2>
<MARQUEE DIRECTION = “DOWN”><H3 STYLE = “COLOR :
GREEN”><CENTER> FINAL CS STUDENTS DETAILS </CENTER> </
MARQUEE><BR><BR></H3><CENTER><TABLE BORDER = “1”>
<THEAD><TR>
<TH> Roll_No </TH><TH> NAME </TH><TH> Marks 1 </TH>
<TH> Marks 2 </TH><TH> Marks 3 </TH><TH> Marks 4 </TH>
<TH> Marks 5 </TH><TH> Total </TH><TH>Percentage </TH>
</TR>
</THEAD>
<TFOOT><TR>
<TH COLSPAN = “4”> STUDENT CATALOG</TH>
</TFOOT>
<TR>
<TD> 001 </TD><TD> RAM </TD>
<TD> Marks 1 : 70 </TD><TD> Marks 2 : 80 </TD>
<TD> Marks 3 : 50 </TD><TD> Marks 4 : 60 </TD>
<TD> Marks 5 : 70 </TD><TD> 330 </TD><TD> 66.0 </TD>
</TR>
</TABLE></CENTER></body></html>
```

Difference :

S. No.	DTD	XSD
1.	DTD is a set of markup declarations that define a document type for an SGML.	XSD specifies how to describe the elements in an XML document formally.
2.	DTD stands for Document Type Definition.	XSD stands for XML Schema Definition.
3.	DTD provides less control over the XML structure.	XSD provides more control over the XML structure.
4.	DTD does not support data types.	XSD supports data types.

DTD is used to define the structure of XML documents.

Que 2.26. **What are XML parsers ? Explain the types of parsers with their advantages and disadvantages.**

AKTU 2018-19, Marks 07

Answer

XML parser :

1. An XML parser is a software library or package that provides interfaces for client applications to work with an XML document.
2. The XML Parser is designed to read the XML and create a way for programs to use XML.
3. XML parser validates the document and check that the document is well formatted.

There are two types of XML Parsers :

1. DOM :

a. The Document Object Model (DOM) is a platform-independent and language-independent standard object model for representing HTML or XML and related formats.

b. The DOM defines a standard for accessing documents like XML and HTML.

Advantages :

i. It supports both read and write operations and the API is very simple to use.

ii. It is preferred when random access to widely separated parts of a document is required.

Disadvantages :

i. It is memory inefficient.

ii. It is comparatively slower than other parsers.

2. SAX :

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

b. SAX is an event-driven API that allows us to interpret a web file that uses XML.

c. SAX takes the control of event specifies by the programmer and handles the situation.

Advantages :

i. It is simple and memory efficient.

ii. It is very fast and works for huge documents.

Disadvantages :

i. It is event based so its API is less intuitive.

ii. Clients never know the full information because the data is broken into pieces.

Que 2.27. "Document Type Definition (DTD) in XML is necessary", justify the statement with suitable example. Under which conditions which DTD is more preferable ?

Answer

"Document Type Definition (DTD) in XML is necessary" :

1. The main purpose of Document Type Definition is to define the structure of an XML document.
2. It contains a list of legal elements and defines the structure with the help of them.
3. DTD provides less control on XML structure.
4. With DTD, independent groups of people can agree to use a common DTD for interchanging data.

Example : Refer Q. 2.22, Page 2–26D, Unit-2.

External DTD is preferable in following conditions :

1. If we want to use same DTD in more than one XML file.
2. When we define DTD for particular kind of data and then use it in different XML file.

Internal DTD is preferable only when we want to use the DTD for a particular XML file only.

PART-6

XML Schemas, Object Model, Presenting and using XML using XML Processors : DOM, and SAX, Dynamic HTML.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 2.28. 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

1. XML schema is an XML based alternative to Document Type Definition (DTD).
2. 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 child 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. 2.21, Page 2–26D, 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 markup declarations that provide a grammar for a class of documents.
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.	**Example :** <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”/>	**Example :** <!DOCTYPE not [<!ELEMENT not (to, from, heading, body)> <!Element to (# PCDATA) > <!Element from (# PCDATA)> <! Element heading (?)> <!Element body (#PCDATA)]>

Que 2.29. **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>
<name> “Web application development with XML and Java”
</name>
</book>
</goods>
</book-order>
```

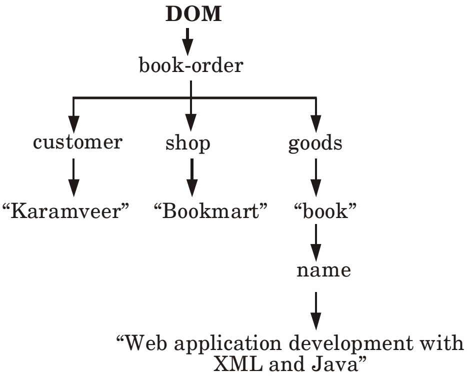


Fig. 2.29.1.

Que 2.30. **Explain COM and DCOM in detail.**

AKTU 2016-17, Marks 10

OR

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 the 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 is a model as COM but it is specially designed for distributed application.
4. DCOM, which originally was called "Network OLE" extends Microsoft's COM, and provides the communication substrate under Microsoft's COM+ application server infrastructure.
5. DCOM was a major competitor to CORBA.

Que 2.31. **What are XML processors ?**

Answer

1. 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.
2. XML processors are written in Java. Some are validating processors, while others are non-validating.
3. When reading an XML document, a validating processor checks the validity constraints and the well-formed constraints defined in XML 1.0 recommendation.
4. A validating XML processor is one of the most robust and faithful implementation of XML processor.
5. The validating processor XML for Java is a Java class library.

Que 2.32. **Explain the term SAX with suitable example.**

Answer

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

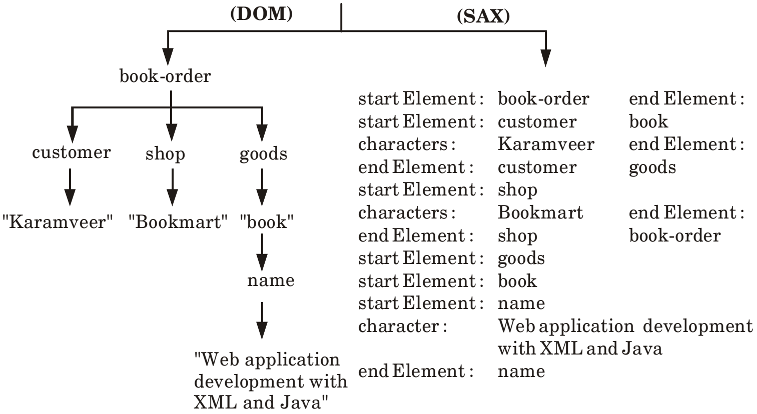


Fig. 2.32.1.

Que 2.33. 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 2.34. 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) and JavaScript.
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.

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 list in HTML ? What are the different types of lists in HTML ? Give an example of each type.

Ans. Refer Q. 2.4.

Q. 2. Explain the table tag with its attributes in detail.

Ans. Refer Q. 2.5.

Q. 3. Explain frames in HTML with example.

Ans. Refer Q. 2.8.

Q. 4. Discuss how frames play a big role in advertising on web. What roles do form play in making web page dynamic ?

Ans. Refer Q. 2.10.

Q. 5. Discuss forms in HTML. Explain various input items used in HTML forms.

Ans. Refer Q. 2.12.

Q. 6. What is CSS ? What are different ways to create them ? Explain with example.

Ans. Refer Q. 2.16.

Q. 7. Describe the role and importance of CSS in web designing. Also differentiate Class and Id in CSS.

Ans. Refer Q. 2.18.

Q. 8. Explain the role of DTD in XML and also describe its types with an example.

Ans. Refer Q. 2.23.

Q. 9. What is XML ? Discuss the significance of XML. How is XML different from HTML ? Explain the process of XML parsing. How are they useful ?

Ans. Refer Q. 2.24.

Q. 10. What are XML parsers ? Explain the types of parsers with their advantages and disadvantages.

Ans. Refer Q. 2.26.

Q. 11. What is XML schema ? Compare XML schema and XML DTD.

Ans. Refer Q. 2.28.

☺☺☺

3
UNIT

Scripting and Networking

CONTENTS

PART-1

Scripting : JavaScript : Introduction, Documents, Forms.

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 ?

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 build 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. 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.

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

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.

Weakness/Disadvantages of JavaScript :

1. There are issues of incompatibility of several scripting that result in website overloading.
2. Different layout engines may render JavaScript differently resulting in inconsistency in terms of functionality and interface.
3. JavaScript is also a common tool for the web hackers; they use scripts injection to hack a site.

4. 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

1. A document object represents the HTML document that is displayed in the window.
2. 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.
3. The hierarchical structure of object is applied for the organization of objects in a web document which include following object :
   a. **Window object :** It is the top most element of the object hierarchy.
   b. **Document object :** Each HTML document that gets loaded into a window becomes a document object which contains the contents of the page.
   c. **Form object :** Everything enclosed in the <form>...</form> tags sets the form object. The form object contains all the elements defined for that object such as text fields, buttons, radio buttons, and checkboxes.
4. When a web page is loaded, the browser creates a Document Object Model of the page.
5. 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 3–3D, Unit-3.

JavaScript function for e-mail address validation :

```
<script type = “text/javascript”>
function validateEmail()
{
        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 ==" "));
        {
        alert("Please Enter your Email ID");
        emailID.focus();
        return false;
        }
        if (str.indexOf(".") == –1 || str.indexOf(".") == 0 ||
        str.indexOf("@") ==–1 || str.indexOf(".") == index_at
        ||str.indexOf("@",(index_at+1))!= –1) ||
        ||str.indexOf(" "))!= –1 ||
        {
            alert("Invalid Email ID");
            return false;
            }
        return true;
    }
</script>
```

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 is 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.

For example :

If a form field (fname) is empty, validateform 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 :

```
<script type=“text/javascript”>
function validateform()
{
var name = document.myform.name.value;
var password = document.myform.password.value;
var confirmpassword = document.myform.password2.value;
var email = document.myform.email.value;
# Username validation
if (name == null  ||  name == “”){
        alert(“Name can’t be blank”);
        return false;
}else if(password.length<6){
        alert(“Password must be at least 6 characters long.”);
        return false;
        }
}
# Retype password validation
if (password == confirmpassword){
return true;
}
else{
alert(“password must be same!”);
return false;
}
}
# Email validation
var emailErr = True;
 if(email == “”) {
        printError(“emailErr”, “Please enter your email address”);
    } else {
       // Regular expression for basic email validation
        var regex = /^\S+@\S+\.\S+$/;
        if(regex.test(email) === false) {
          printError(“emailErr”, “Please enter a valid email address”);
        } else{
                printError(“emailErr”, “”);
                emailErr = false;
                }
        }
</script>
<body>
<form name=“myform” method=“post” onsubmit=“return validateform()”>
Username:   <input type=“text” name=“name”><br/>
```

```
Password : <input type=“password” name=“password” /><br/>
Re-enter Password: <input type=“password” name=“password2”/><br/>
Email : <input type=“text” name=“email”>
<div class=“error” id=“emailErr”></div>
<input type=“submit” value=“Register”>
</form>
</body>
</html>
```

Que 3.7. **Compare Java and JavaScript. Write a JavaScript program to define a user defined function for sorting the values in an array.**

AKTU 2018-19, Marks 07

Answer

Comparison : Refer Q. 3.2, Page 3–3D, Unit-3.

Program to sort value using JavaScript :

```
<!DOCTYPE html>
<html><body>
<h2>JavaScript Array Sort</h2>
<p>The sort() method sorts an array alphabetically.</p>
<button onclick=“myFunction()”>Try it</button>
<p id=“demo”></p>
<script>
var fruits = [“Banana“, “Orange”, “Apple”, “Mango”];
document.getElementById(“demo”).innerHTML = fruits;
function myFunction() {
fruits.sort();
document.getElementById(“demo”).innerHTML = fruits;
}
</script></body></html>
```

Que 3.8. **Create an html page named as “String_Math.html” and within the script tag define some string variables and use different string function to demonstrate the use of the predefined functions. Do the same for the math function.**

AKTU 2018-19, Marks 07

Answer

```
<html><body>
<script type="text/javascript">
var str1="W3Schools is great!"
document.write(str1.substr(2,6))
document.write("<br><br>")
document.write(str1.substring(2,6))
// string length
var str2="Web Enabling Tools is Cool!"
document.write("<p>" + str2 + "</p>")
document.write("str.length")
// Square root of a number
var property_value = Math.SQRT1_2
document.write("Property Value is : " + property_value);
var value = Math.exp(1);
document.write("First Test Value : " + value );
var value = Math.exp(30);
document.write("<br />Second Test Value : " + value );
var value = Math.exp(– 1);
document.write("<br />Third Test Value : " + value );
// Find maximum of a number
var value = Math.max(10, 20, – 1, 100);
document.write("First Test Value : " + value );
var value = Math.max(– 1, – 3, – 40);
document.write("<br />Second Test Value : " + value );
</script>
```

Que 3.9. **Explain the role of JavaScript to develop a web page. Write a JavaScript function to check a textbox is either empty or not.**

Answer

Role of JavaScript :

1. It makes our website dynamic.
2. It makes our webpage interactive which means that it can respond to mouse clicks, double clicks, hover and many other actions (called events).

3. It can also be used for modifying our html, content and styles, form validation.
4. JavaScript can interact with the website server to send and receive information to update UI in real-time.

JavaScript function to check whether a textbox is either empty or not :

```
<html>
<script>
function required(inputtx)
{
if (inputtext1.value.length == 0)
{
alert("Textbox is empty");
return false;
}
return true;
}
</script>
<body>
<form>
<input type = "text" name="text1">
</form>
</body></html>
```

PART-2

Statements, Functions, Objects.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 3.10. 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
}
```

For 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
}
```

For 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 with different condition.

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
}
```

For 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
}
```

For 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 3.11. **Describe data types and functions in JavaScript.**

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
}
```

var1, var2, etc., are variables or values passed into the function.

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

```
function function_name ()
{
```

```
some code
}
```

Que 3.12. **Compare Java and JavaScript. Explain and demonstrate 5 different types of objects in JavaScript with example.**

AKTU 2019-20, Marks 07

Answer

Comparison : Refer Q. 3.2, Page 3–3D, Unit-3.

Different object 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)
{
        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.

For example :

```
<html>
<head>
<title>JavaScript Object </title>
</head>
<body>
<script type="text/javascript">
var value = Math.acos(-1);
document.write("ACOS Test Value : " + value +"<br>");
// Math object
var d = new Date();
document.write("<b>Locale String:</b> " +
d.toLocaleString()+"<br>");
// Date Object
var str = "CareerRide Info"; //String Object
var s = str.split();
document.write("<b>Char At:</b> " + str.charAt(1)+"<br>");
document.write("<b>CharCode At:</b> " + str.charCodeAt(2)+"<br>");
document.write("<b>Index of:</b> " + str.indexOf("ide")+"<br>");
</script>
</body>
</html>
```

Output:

```
ACOS Test Value : 3.141592653589793
Locale String: 22/02/2020 11:14:06 AM
Char At: a
```

CharCode At: 114

Index of: 7

Que 3.13. **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.

   For 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>
```

Que 3.14. **How do we handle event in JavaScript ? What is DHTML ?**

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. 3.14.1 depicts how events occur and are handled in JavaScript.

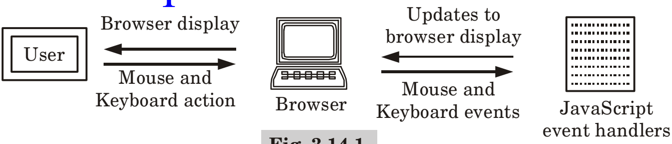

User
Browser display
Mouse and Keyboard action
Browser
Updates to browser display
Mouse and Keyboard events
JavaScript event handlers

Fig. 3.14.1.

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. In JavaScript, we can change the default action and write our own event handler that will be associated with the 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. 2.34, Page 2–39D, Unit-2.

PART-3

Introduction to AJAX.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 3.15. **Write a short note on AJAX.**

AKTU 2016-17, Marks 05

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. 3.15.1.
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.

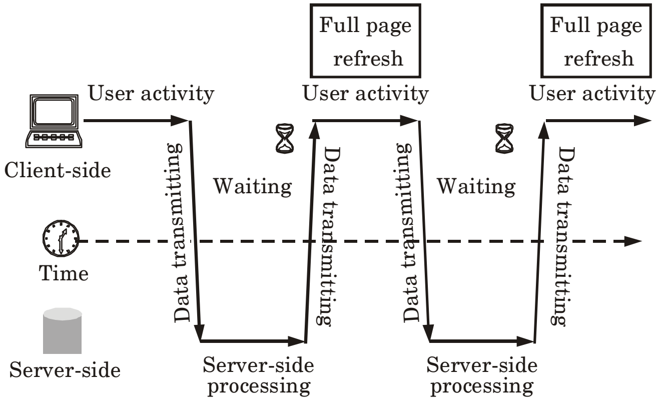


Fig. 3.15.1. Traditional web application communication flow.

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. 3.15.2.

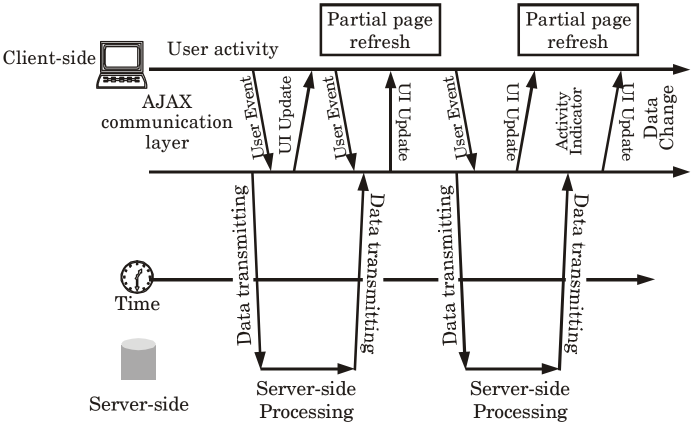


Fig. 3.15.2. AJAX style communication flow.

Que 3.16. **What is AJAX ? Explain its advantage and its working. Explain with example.**

AKTU 2017-18, Marks 10

Answer

AJAX : Refer Q. 3.15, Page 3–18D, Unit-3.

Advantages of AJAX :

1. Reduces the server traffic and increases the speed.
2. Ajax is responsive and time for data transfer is also less.
3. Form validation
4. Bandwidth usage can be reduced.
5. Asynchronous calls can be made which reduces the time for data arrival.

Working of AJAX : XMLHttpRequest object plays an important role as AJAX communicates with the server using XMLHttpRequest object.

1. User sends a request from the UI and a JavaScript call goes to XMLHttpRequest object.
2. HTTP request is sent to the server by XMLHttpRequest object.
3. Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
4. Data is retrieved.
5. Server sends XML data or JSON data to the XMLHttpRequest callback function.
6. HTML and CSS data is displayed on the browser.

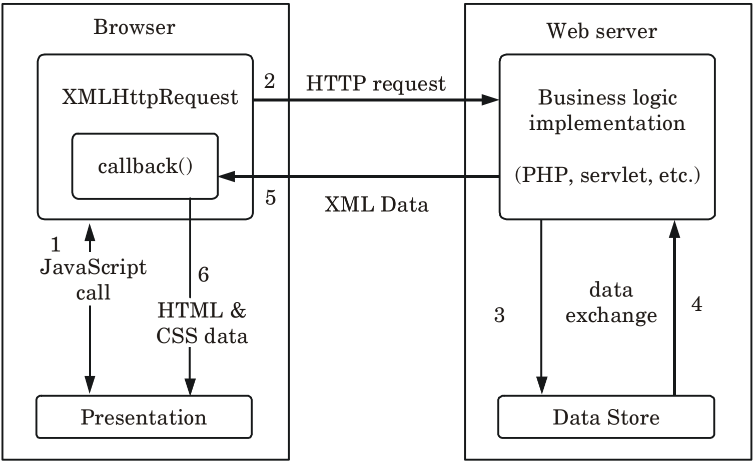


Fig. 3.16.1.

For 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 3.17. **What is AJAX ? Explain the application of AJAX with the help of suitable examples.** **AKTU 2015-16, Marks 10**

OR

Discuss AJAX. Explain the application of AJAX with the help of suitable examples. **AKTU 2019-20, Marks 07**

Answer

AJAX : Refer Q. 3.15, Page 3–18D, 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.

4. Major application of AJAX is in login forms where user can enter their login details directly on the original page.

For example :

```
<!DOCTYPE html>
<html>
<style>
table,th,td {
    border : 1px solid black;
    border-collapse: collapse;
}
th,td {
    padding: 5px;
}
</style>
<body>
<h1>The XMLHttpRequest Object</h1>
<form action="">
<select name="customers" onchange="showCustomer(this.value)">
<option value="">Select a customer:</option>
<option value="name1">Aakash Pandey</option>
<option value="name2">Mohan</option>
<option value="name3">Harshit Kumar</option>
</select>
</form>
<br>
<div id="txtHint">Customer info will be listed here...</div>
<script>
function showCustomer(str) {
     var xhttp;
    if (str == "") {
    document.getElementById("txtHint").innerHTML = "";
    return;
}
xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
   if (this.readyState == 4 && this.status == 200) {
       document.getElementById("txtHint").innerHTML = this.responseText;
       }    };
xhttp.open("GET", "getcustomer.asp?q="+str, true);
xhttp.send();
}
</script>
</body>
</html>
```

PART-4

Networking : Internet Addressing, InetAddress, Factory Methods, Instance Methods.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 3.18. Explain IP addressing.

Answer

1. The IP address is a network layer address that uniquely identifies each computer on network.
2. Each TCP/IP host is identified by a logical IP address.
3. The IP address identifies a system's location on the network. An IP address must be globally unique and have a uniform format.
4. Each IP address includes a network ID and a host ID.
   i. 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.
   ii. 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.
5. 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. 3.18.1.

6. 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.
7. 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 3.19. **Give the classification of different IP address.**

Answer

IP address is classified as :

1. **Class *A* :**
   i. Class *A* addresses are assigned to networks with a very large number of hosts.
   ii. The high-order bit in a class *A* address is always set to zero.
   iii. The next seven bits (completing the first octet) complete network ID. The remaining 24 bits (the last three octets) represent the host ID.
   iv. This allows for 126 networks and 16,777,214 *i.e.*, 2^{24} hosts per network.
2. **Class *B* :**
   i. Class *B* addresses are assigned to medium-sized to large-sized networks.
   ii. The two high-order bit in a class *B* address are always set to binary 10.
   iii. The next 14 bits (completing the first two octets) complete the network ID. The remaining 16 bits (last two octets) represent the host ID.
   iv. This allows for 16,384 networks and 65,534 hosts per network.

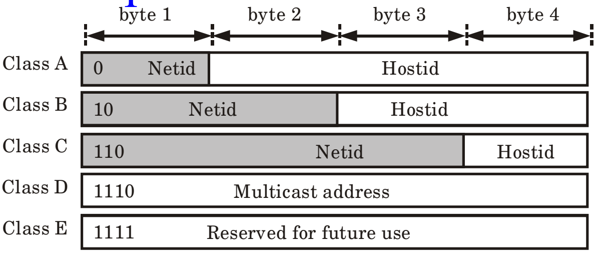


Fig. 3.19.1.

3. **Class *C* :**
   i. Class *C* addresses are used for small networks.
   ii. 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.

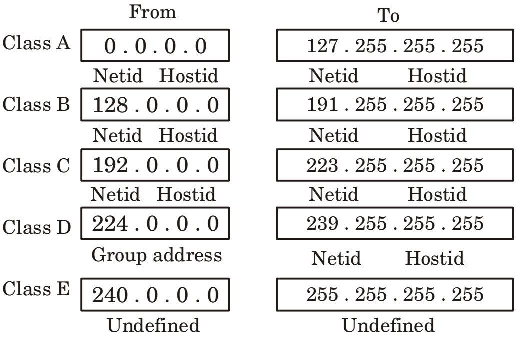


Fig. 3.19.2.

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 3.20. **What is InetAddress class ? Explain factory method of InetAddress.**

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 IntAddress object.
2. Factory method is a static method in a class and return an instance of that class.
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.

Que 3.21. **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.
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 :
   - **a.** **Boolean equals(Object other) :** It returns true if this object has the same Internet address as other. Otherwise, it returns false.
   - **b.** **Byte[ ] getAddress() :** It returns a byte array that represents the object's Internet address in network byte order.
   - **c.** **String getHostAddress() :** It returns a string that represents the host address associated with the InetAddress object.

d. **String getHostName() :** It returns a string that represents the host name associated with the InetAddress object.

e. **String toString() :** It returns a string that lists the host name and the IP address.

PART-5

TCP/IP Client Sockets, URL, URL Connection, TCP/IP Server Socket, Datagram.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 3.22. Explain TCP/IP client socket. Also, write the constructor and methods used to create a client socket.

Answer

1. TCP/IP client sockets are used to implement bi-directional, point-to-point, stream-based connections between hosts on the Internet.
2. 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.
3. The creation of a Socket object implicitly establishes a connection between the client and server.
4. Following are the two constructors used to create client socket :

   a. **Socket(String hostName, int port) :** Creates a socket connecting the local host to the named host, port and can throw an Unknown HostException or an IOException.

   b. **Socket(InetAddress ipAddress, int port) :** Creates a socket using a pre-existing InetAddress object, a port and can throw an IOException.

5. Following methods are used by TCP/IP client socket :

   a. **InetAddress getInetAddress() :** Returns the InetAddress associated with the Socket object.

   b. **Int getPort() :** Returns the remote port to which the Socket object is connected.

   c. **Int getLocalPort() :** Returns the local port to which the Socket object is connected.

Que 3.23. **Write short note on URL and URLConnection class in Java.**

Answer

1. URL is an acronym for Uniform Resource Locator.
2. It points to a resource on the World Wide Web (WWW).
3. A URL contains many information like protocol name, server name, port number and file name.
4. The URL is represented by an URL class.
5. Consider the following URL :

   http://www.quantumpage.com/aktu-paper.html

   a. **Protocol :** In this case, http is the protocol.

   b. **Server name or IP address :** In this case, www.quantumpage.com is the server name.

   c. **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.

   d. **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 :

1. The Java URLConnection class represents a communication link between the URL and the application.
2. This class can be used to read and write data to the specified resource referred by the URL.

3. The openConnection() method of URL class returns the object of URLConnection class.
4. Syntax to get the object of URLConnection :

   public URLConnection openConnection()throws IOException{}
5. The URLConnection class use 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 3.24. **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 3.25. **Explain the following terms in brief :**

i. Socket programming
ii. TCP/IP Server

Answer

i. Socket programming :

1. Java socket programming is used for communication between the applications running on different JRE.
2. Java socket programming can be connection-oriented or connection-less.
3. Socket and ServerSocket classes are used for connection-oriented socket programming and DatagramSocket and DatagramPacket classes are used for connection-less socket programming.

4. Sockets provide the communication mechanism between two computers using TCP.
5. A client program creates a socket on its end of the communication and attempts to connect that socket to a server.
6. The client in socket programming must know :

a. IP address of server

b. Port number

ii. TCP/IP server : Refer Q. 3.24, Page 3–29D, Unit-3.

Que 3.26. 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 datagram between applications.
2. Datagram socket is a type of network socket which provide connectionless point for sending and receiving packets.
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.

Que 3.27. **Discuss socket and server socket in Java with its package. Write a program in Java to demonstrate, how the communication is establish between client and server ?**

AKTU 2019-20, Marks 07

Answer

Socket : Refer Q. 3.22, Page 3–27D, Unit-3.

Server socket : Refer Q. 3.24, Page 3–29D, Unit-3.

For example :

```
// A Java program for a Client
import java.net.*;
import java.io.*;
public class Client
{
// initialize socket and input output streams
private Socket socket = null;
private DataInputStream input = null;
private DataOutputStream out = null;
// constructor to put ip address and port
public Client(String address, int port)
{
// establish a connection
try
{
socket = new Socket(address, port);
System.out.println("Connected");
// takes input from terminal
input = new DataInputStream(System.in);
// sends output to the socket
out = new DataOutputStream(socket.getOutputStream());
}
catch(UnknownHostException u)
{
System.out.println(u);
}
catch(IOException i)
{
System.out.println(i);
}
// string to read message from input
String line = “”;
// keep reading until “Over” is input
while (!line.equals(“Over”))
{
try
{
line = input.readLine();
out.writeUTF(line);
}
catch(IOException i)
{
System.out.println(i);
}
}
// close the connection
try
{
input.close();
out.close();
socket.close();
}
catch(IOException i)
{
System.out.println(i);
}
}
public static void main(String args[ ])
{
Client client = new Client(“127.0.0.1”, 5000);
}
}
```

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 ? What are the features of JavaScript ?

Ans. Refer Q. 3.1.

Q. 2. What is the difference between Java and JavaScript ? Describe the strengths and weakness of JavaScript.

Ans. Refer Q. 3.2.

Q. 3. What are scripting languages and why JavaScript is used ? Write a JavaScript function for validating form data like mandatory fields and email field.

Ans. Refer Q. 3.6.

Q. 4. Explain the role of JavaScript to develop a web page. Write a JavaScript function to check a textbox is either empty or not.

Ans. Refer Q. 3.9.

Q. 5. Explain conditional statements used in JavaScript with example.

Ans. Refer Q. 3.10.

Q. 6. Compare Java and JavaScript. Explain and demonstrate 5 different types of objects in JavaScript with example.

Ans. Refer Q. 3.12.

Q. 7. What is AJAX ? Explain its advantage and its working. Explain with example.

Ans. Refer Q. 3.16.

Q. 8. What is AJAX ? Explain the application of AJAX with the help of suitable examples.

Ans. Refer Q. 3.17.

Q. 9. Explain TCP/IP client socket. Also, write the constructor and methods used to create a client socket.

Ans. Refer Q. 3.22.

Q. 10. What is datagram ? Give its characteristics. Also, explain datagram socket.

Ans. Refer Q. 3.26.

☺☺☺

4
UNIT

Enterprise Java Beans and JDBC

CONTENTS

PART-1

Enterprise Java Bean : Preparing a Class to be a Java Beans, Creating a Java Beans, Java Beans Properties.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

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

Answer

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

Advantages of EJB :

1. It can run in multithreaded environment.
2. It contains only business logic.
3. EJB provides distributed transaction support.
4. It provides portable and scalable solutions of the problem.
5. It provides a mechanism to store and retrieve data in a persistent way.

Disadvantages of EJB :

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

Que 4.2. **Discuss EJB. Explain EJB architecture. What are its various types ?**

AKTU 2019-20, Marks 07

OR

Write short note on EJB architecture. **AKTU 2016-17, Marks 05**

Answer

EJB : Refer Q. 4.1, Page 4–2D, Unit-4.

EJB architecture :

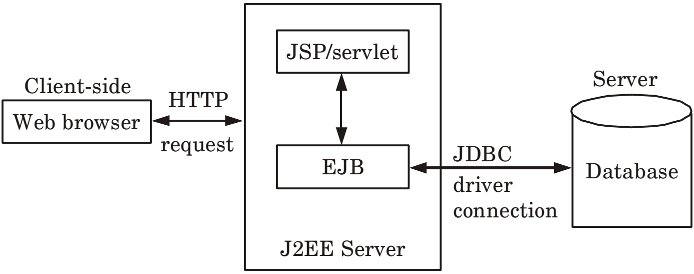


Fig. 4.2.1. Architecture.

The EJB architecture is an extension of web architecture.

Working of EJB architecture :

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

Types of EJB :

1. **Entity bean :** Entity beans represent persistent data storage. Entity beans are used for modeling the business concept.
2. **Session bean :** Session beans are used for managing processes or tasks. Hence, session beans are used for managing activities.
3. **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.

Que 4.3. **What is Java Bean exactly ? Write down the steps to create Java Bean. What is the role of introspection in Java Bean ?**

OR

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

Answer

EJB architecture and its types : Refer Q. 4.2, Page 4–2D, Unit-4.

Java Beans :

1. Java Beans are classes which encapsulate several objects into a single object.
2. It helps in accessing the objects from multiple places.
3. It is a portable, platform independent model written in Java.

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 SimpleBean.jar".

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.
2. Select SimpleBean.jar.
3. 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 java.awt.Canvas.

Role of introspection in Java Bean :

1. Introspection in Java is used in the context of Java Beans which defines the component model of Java.
2. Introspection feature enables a Java Bean to get the properties, methods and events of other beans at runtime.
3. This helps the developers to design and develop their beans without knowing the details of other beans.

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 4.4. Explain JavaBeans. Why they are used ? Discuss setter and getter method with Java code. **AKTU 2019-20, Marks 07**

Answer

Java Beans : Refer Q. 4.3, Page 4–3D, Unit-4.

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 constructor with no argument.
4. It can be manipulated visually in a builder tools.

Getter and setter method :

1. In Java, getter and setter are two conventional methods that are used for retrieving and updating value of a variable.

For example :

The following code is an example of simple class with a private variable and a couple of getter/setter methods :

```
public class SimpleGetterAndSetter {
private int number;
public int getNumber() {
return this.number;
}
public void setNumber(int num) {
this.number = num;
}
}
```

2. The class declares a private variable, number. Since "number" is private, code from outside this class cannot access the variable directly,

```
SimpleGetterAndSetter obj = new SimpleGetterAndSetter();
obj.number = 10;
int num = obj.number;
```

3. Inside the main class invoke the getter *i.e.*, getNumber() and the setter *i.e.*, setNumber() in order to read or update the variable,

   For example :

```
public static void main (String args[ ])
{
SimpleGetterAndSetter obj = new SimpleGetterAndSetter();
obj.setNumber(10);
System.out.println (obj.getNumber ( ));
}
```

4. Getter and setter are also known as accessor and mutator in Java.

Que 4.5. How to prepare a class to be a Java Beans ?

Answer

Java Beans 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 getter 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 4.6. **Explain Java Bean class properties.**

Answer

Java Bean class contains three types of properties :

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.
2. **Boolean properties :**
   a. A boolean property has a value of true or false.
   b. 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();

c. For getting the values isN and getN methods are used and for setting the Boolean values setN method is used.

3. **Indexed properties :**

a. An indexed property consists of multiple values.

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(int index);

public void setN(int index, T value);

public T[ ] getN( );

public void setN(T values[ ]);

PART-2

Types of Beans, Stateful Session Bean, Stateless Session Bean, Enlity Bean.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 4.7. **Explain session beans with its types.**

Answer

Session bean :

1. Session bean encapsulates business logic only, it can be invoked by local, remote and web service client.
2. It can be used for managing activities like database access, calculation etc.
3. The life cycle of session bean is maintained by the application server (EJB container).
4. Session bean is created by a customer and its duration is only for the signal client server session.

Types of session bean :

1. **Stateless session bean :**

a. Stateless session bean is a business object that represents business logic only. It does not have state (data).

b. The stateless bean objects are pooled by the EJB container to service the request on demand.

c. It can be accessed by one client at a time.

d. The stateless session bean is distributed object which has no connection with informal state; only allow parallel access to beans.

e. Annotations used in stateless session bean are :

i. @Stateless

ii. @PostConstruct

iii. @PreDestroy

2. **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

3. **Singleton session beans :**

a. A singleton session bean is instantiated once per application and exists for the lifecycle of the application.

b. Singleton session beans are designed for circumstances in which a single enterprise bean instance is shared across and concurrently accessed by clients.

c. It has only one singleton session bean per application.

d. It can implement web service endpoints.

e. Singleton session beans maintain their state between client invocations but are not required to maintain their state across server crashes or shutdowns.

Que 4.8. **Describe entity beans with its types.**

Answer

1. Entity beans are objects that represent a persistence storage mechanism.

2. Each entity beans has underlying table in a relational database and each row in the table represents the instance of the bean.

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.

Que 4.9. **What do you mean by session bean ? Explain its types using suitable example.**

Answer

Session bean and its types : Refer Q. 4.7, Page 4–8D, Unit-4.

Example : An example of stateless session bean is a stock quote component that returns the current price of a given stock symbol. Such a bean could look up the stock price from a database that is updated by a real-time feed.

An example of a stateful session bean is a shopping cart that represents the collection of products selected by a particular customer for purchase during a session. The shopping cart should not be shared because it represents a particular interaction with a particular customer and is alive only for the customer's session.

PART-3

Java Database Connectivity (JDBC) : Merging Data from Multiple Tables : Joining.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 4.10. **What is JDBC ? How it works ?**

Answer

1. JDBC (Java Database Connectivity) is a Java API that manages connection to database, issuing queries and commands and handling result sets obtained from the database.
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.

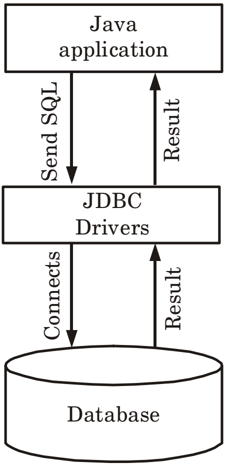


Fig. 4.10.1.

Que 4.11. 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 4.12. **Explain JDBC application architecture.**

Answer

JDBC architecture :

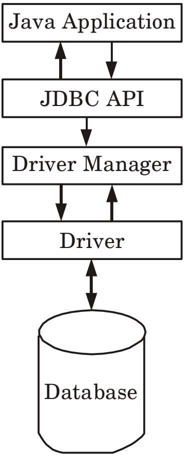


Fig. 4.12.1.

1. **Java application :** It is a standalone Java program which uses the JDBC API to get connected and perform operations on the database data.
2. **JDBC API :** It is a set of classes and interfaces used in a Java program for database operations. Java.sql and Javax.sql packages provide the necessary library support.
3. **Driver manager :** Java program uses DriverManager class to get the connection with the database.
4. **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.
5. **Database :** It is a collection of all enterprise data.

Que 4.13. Explain the types of JDBC drivers.

Answer

Types of JDBC drivers are :

1. **JDBC-ODBC bridge driver (Type 1 driver) :**
   a. These drivers are the bridge drivers such as JDBC-ODBC bridge.
   b. These drivers rely on an intermediary such as ODBC to transfer the SQL calls to the database.
   c. Bridge drivers often rely on native code, although the JDBC-ODBC library native code is part of the Java-2 virtual machine.

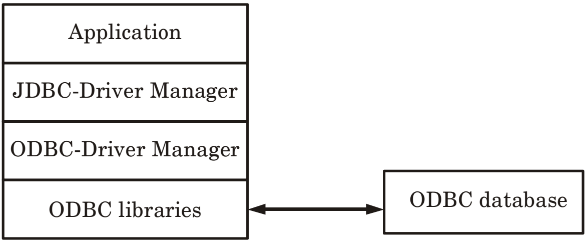


Fig. 4.13.1. JDBC-ODBC bridge driver.

2. **Native API partly Java driver (Type 2 driver) :**
   a. A native API is partly a Java driver. It uses native C language library calls to translate JDBC to native client library.
   b. These drivers are available for Oracle, Sybase, DB2 and other client library based RDBMS.
   c. Type 2 drivers use native code and require additional permission to work in an Applet.

d. A Type 2 driver might need client-side database code to connect over the network.

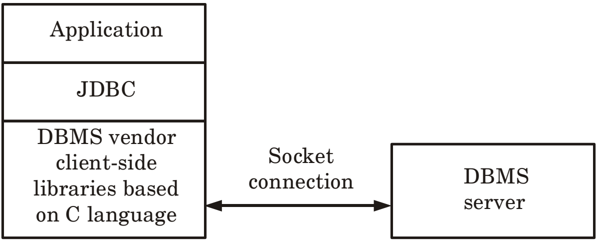


Fig. 4.13.2. Native API partly Java driver.

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

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

   b. Here the calls are translated and sent to middle tier server through the socket.

   c. The middle tier contacts the database.

   d. Type 3 drivers call the database API on the server.

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

   a. A native protocol Java driver contains JDBC calls that are converted directly to the network protocol used by the DBMS server.

   b. This driver interacts directly with database.

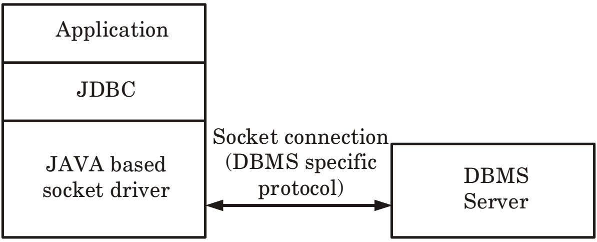


Fig. 4.13.3. Native protocol pure Java driver.

   c. It does not require any native database library. So, it is also called thin driver.

Que 4.14. **What is JDBC ? Explain the drivers used in JDBC. Write a JDBC program for insert and display the record of employees using prepared statement.** **AKTU 2018-19, Marks 07**

Answer

JDBC : Refer Q. 4.10, Page 4–10D, Unit-4.

Drivers in JDBC : Refer Q. 4.13, Page 4–13D, Unit-4.

Program :

```
import java.sql.*; import java.io.*;
public class PreparedStatementDemo1 {
Connection con;
PreparedStatement ps;
public PreparedStatementDemo1() {
try {
Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost/
test?user=root&password=root");
} catch (Exception e) { e.printStackTrace(); } }
// add customer detail
public String addCustomer(String custid, String name, String address,
String contact) {
String status = "";
try {
ps = con.prepareStatement("insert into Customer values(?,?,?,?)");
ps.setString(1, custid); ps.setString(2, name);
ps.setString(3, address); ps.setString(4, contact);
int i = ps.executeUpdate();
if (i != 0) {
status = "Inserted";
} else {
status = "Not Inserted"; }
} catch (Exception e) { e.printStackTrace(); }
return status; }
// customer record
public void searchCustomer(String custid) {
String sql = "";
if (custid.trim().length() == 0) {
sql = "select * from Customer";
} else {
sql = "select * from Customer where custid=" + custid + ""; }
try {
ps = con.prepareStatement(sql);
ResultSet res = ps.executeQuery();
while (res.next()) {
System.out.print(res.getString(1));
System.out.print(res.getString(2));
System.out.print(res.getString(3));
System.out.println(res.getString(4)); }
} catch (SQLException e) {e.printStackTrace(); } }
```

```
public String deleteCustomer(String custId) {
String status = “”;
try {
ps = con.prepareStatement(“delete from Customer where custid=?”);
ps.setString(1, custId);
int i = ps.executeUpdate();
if (i != 0) {
status = “Customer details deleted”;
} else {
status = “Customer details not deleted”; }
} catch (Exception e) {e.printStackTrace(); }
return status; }
public void menuDisplay() {
try {
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
int ch = 0;
while (true) {
System.out.println(“== Customer Management System == \n”
+ “1. Add Customer \n” + “2. Display Customer’s record \n”
+ “3. Exit \n” + “Enter Choice \n”);
String str1 = br.readLine().toString();
ch = Integer.parseInt(str1);
switch (ch) {
case 1: {
System.out.println(“Enter Customer Id”);
String custId = br.readLine();
System.out.println(“Enter Customer Name”);
String custName = br.readLine();
System.out.println(“Enter  Customer Address”);
String custAddress = br.readLine();
System.out.println(“Enter Customer Contact No.”);
String custContact = br.readLine();
System.out.println(addCustomer(custId, custName,
custAddress, custContact));
break; }
case : 2{
System.out.println(“Enter Customer Code to display record”);
String custId = br.readLine();
searchCustomer(custId);
break; }
case 3: {
System.exit(0); }
default:
break; } }
} catch (Exception e) {
e.printStackTrace(); } }
```

```
        public static void main(String[] args) {
        PreparedStatementDemo1 obj = new
        PreparedStatementDemo1();
        obj.menuDisplay(); } }
```

PART-4

Manipulating, Databases with JDBC, Prepared Statements Transaction Processing, Stored Procedures.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 4.15. **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 4.16. **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. 4.12, Page 4–12D, Unit-4.
Types of JDBC driver : Refer Q. 4.13, Page 4–13D, 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 4.17. **Explain PreparedStatement interface in JDBC.**

Answer

1. Prepared statement interface is a subinterface of statement.
2. It is used to execute parameterized query.
3. The PreparedStatement interfaces define the methods and properties that enable us to send SQL or PL/SQL commands and receive data from our database.
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 4.18. **Explain the steps to connect a Java application with database using JDBC.**

Answer

Steps to connect a Java application with 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 :**
   a. createStatement() method is invoked on current connection object to create a SQL Statement.
   b. Syntax :

      public Statement createStatement() throws SQLException

      Example :

      Statement s=con.createStatement();

4. **Execute SQL statement :**
   a. executeQuery() method of statement interface is used to execute SQL statements.
   b. 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 :**
   a. After executing SQL statement, to close the connection and release the session.
   b. The close() method of connection interface is used to close the connection.
   c. Syntax :

      public void close() throws SQLException

      Example :

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

Que 4.19. What do you mean by database drivers, explain each type ? Also explain the steps to get any value into database.

Answer

1. A database driver is a computer program that implements a protocol (ODBC or JDBC) for a database connection.
2. The driver works like an adaptor which connects a generic interface to a specific database vendor implementation.

Types of database drivers : Refer Q. 4.13, Page 4–13D, Unit-4.

Steps to get any value into database : Refer Q. 4.14, Page 4–14D, Unit-4.

Que 4.20. Write a short note on stored procedure in Java.

Answer

1. A program which contains *n* number of SQL statements and residing a database environment is known as stored procedure.
2. Stored procedures are divided into two types :
   a. **Procedure :**
      i. A procedure is one which contains block of statements which will return either zero or more than one value.

ii. **Syntax for creating a procedure :**

```
create procedure <procedure name> (parameters)
as/is
    local variables;
begin
    block of statements;
end;
```

b. **Function :**

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

ii. 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 4.21. **Write the difference between stored procedure and functions.**

Answer

Differences :

S. No.	Stored procedure	Function
1.	It is used to perform business logic.	It is used to perform calculation.
2.	It may or may 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 4.22. **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.

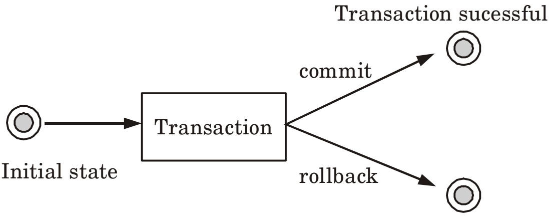


Fig. 4.22.1.

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 4.23. **Describe the transaction management method in JDBC with example.**

Answer

In JDBC, Connection interface provides different methods to manage transaction.

Method	Description
void setAutoCommit(boolean status)	transaction is committed by default.
void commit()	commits the transaction.
void rollback()	cancels the transaction.

For 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=‘Ghaziabad’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
```

VERY IMPORTANT QUESTIONS

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

Q. 1. Discuss EJB. Explain EJB architecture. What are its various types ?

Ans. Refer Q. 4.2.

Q. 2. What is Java Bean exactly ? Write down the steps to create Java Bean. What is the role of introspection in Java Bean ?

Ans. Refer Q. 4.3.

Q. 3. Explain JavaBeans. Why they are used ? Discuss setter and getter method with Java code.

Ans. Refer Q. 4.4.

Q. 4. Explain session beans with its types.

Ans. Refer Q. 4.7.

Q. 5. What is JDBC ? How it works ?

Ans. Refer Q. 4.10.

Q. 6. Explain the types of JDBC drivers.

Ans. Refer Q. 4.13.

Q. 7. Explain JDBC application architecture. What are the various types of JDBC drivers ? Write steps to connect database with the web application using JDBC.

Ans. Refer Q. 4.16.

☺☺☺

5

UNIT

Servlets and Java Server Pages (JSP)

CONTENTS

PART-1

Servlets : Servlet Overview and Architecture, Interface Servlet and the Servlet Life Cycle.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

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

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.

Life cycle of Servlet :

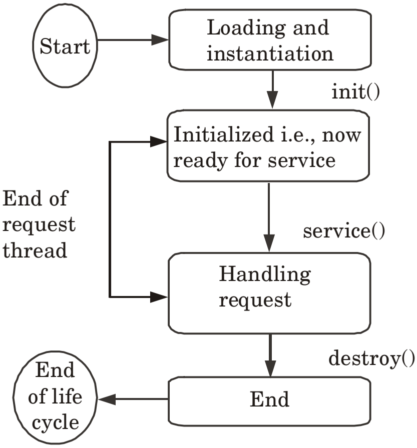


Fig. 5.1.1.

Stages of the Servlet life cycle :

1. **Loading a Servlet :**
   a. The first stage of the Servlet lifecycle involves loading and initializing the Servlet by the Servlet container.
   b. The Servlet container performs two operations in this stage :
      i. **Loading :** Loads the Servlet class.
      ii. **Instantiation :** Creates an instance of the Servlet. To create a new instance of the Servlet, the container uses the no-argument constructor.
2. **Initializing a Servlet :**
   a. After the Servlet is instantiated successfully, the Servlet container initializes the instantiated Servlet object.
   b. The container initializes the Servlet object by invoking the Servlet.init(ServletConfig) method which accepts ServletConfig object reference as parameter.
3. **Handling request :**
   a. After initialization, the Servlet instance is ready to serve the client requests.
   b. The Servlet container performs the following operations when the Servlet instance is located to service a request :
      i. It creates the ServletRequest and ServletResponse objects.
      ii. After creating the request and response objects it invokes the Servlet.service(ServletRequest, ServletResponse) method by passing the request and response objects.
4. **Destroying a Servlet :**
   a. When a Servlet container decides to destroy the Servlet, it performs the following operations,
      i. It allows all the threads currently running in the service method of the Servlet instance to complete their jobs and get released.
      ii. After currently running threads have completed their jobs, the Servlet container calls the destroy() method on the Servlet instance.
   b. After the destroy() method is executed, the Servlet container releases all the references of this Servlet instance so that it becomes eligible for garbage collection.

Characteristics of servlet :

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

Que 5.2. **Explain the life cycle of servlet. Also write a servlet for displaying a string "HELLO WORLD!"** **AKTU 2017-18, Marks 10**

Answer

Life cycle of servlet : Refer Q. 5.1, Page 5–2D, Unit-5.

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 5.3. **What do you mean by Common Gateway Interface (CGI) ? How does CGI work ?**

Answer

1. CGI is an acronym for the Common Gateway Interface which is a standard protocol for running programs within a web server.

2. 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.
3. CGI programs are dynamic and the state of their variables alters as they execute.
4. CGI allows a WWW server to provide information to WWW clients that would otherwise not be available to those clients.

Working of CGI :

1. A reader sends a URL that causes the AOLserver to use CGI to run a program.
2. 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.
3. 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.

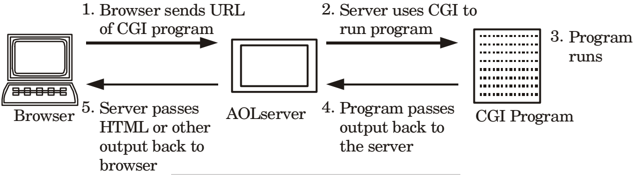

1. Browser sends URL of CGI program
2. Server uses CGI to run program
3. Program runs
Browser
5. Server passes HTML or other output back to browser
AOLserver
4. Program passes output back to the server
CGI Program

Fig. 5.3.1. Working of CGI.

Que 5.4. **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 5–2D, Unit-5.

Java servlets have following advantage over CGI and other API's :

i. **Platform independence :** Java servlets are pure Java program, so it is platform independent. It can run on any servlet enabled web server.

ii. **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.

iii. **Extensibility :** Java servlets are developed in Java which is robust, well-designed and object-oriented language which can be extended into new objects.

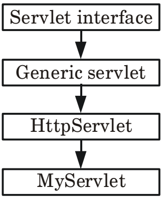
iv. **Safety :** Java servlet provides a very good safety features like memory management, exception handling features and emerged as a very powerful web server extension.

v. **Secure :** Servlets are server-side components, so it inherits the security provided by the web server.

Que 5.5. **Explain servlet architecture.**

Answer

Servlet architecture includes :



1. **Servlet interface :**
   a. To write a servlet we need to implement Servlet interface.
   b. Servlet interface can be implemented directly or indirectly by extending GenericServlet or HttpServlet class.
2. **GenericServlet :**
   a. GenericServlet provides simple versions of the lifecycle methods init( ) and destroy( ).
   b. It also implements the log method which is declared in the ServletContext interface.
3. **HttpServlet :**
   a. HttpServlet is an abstract class present in javax.servlet.http package and has no abstract methods.
   b. It extends GenericServlet class.
   c. 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 5.6. **Write short notes on servlet interface.**

Answer

1. Servlet interface provides common behaviour to all the servlets.
2. Servlet interface needs to be implemented for creating any servlet (either directly or indirectly).
3. Servlet interface defines methods that are implemented by all servlets.
4. 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 ServletConfig.
5.	public String getServletInfo()	It returns information about servlet such as writer, copyright, version etc

Que 5.7. **Explain how servlet works.**

Answer

1. When the web server starts, the servlet container deploys and loads all the servlets.
2. 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.
2. Once the servlet is loaded, the servlet container creates the instance of servlet class. For each instantiated servlet, its init() method is invoked.
3. Client (user browser) sends an Http request to web server on a certain port.
4. Then, the servlet container creates HttpServletRequest and HttpServletResponse objects.
5. 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.
6. The servlet container produce a new thread that calls service() method for each client request.
7. The service() method dispatches the request to the correct handler method based on the type of request.

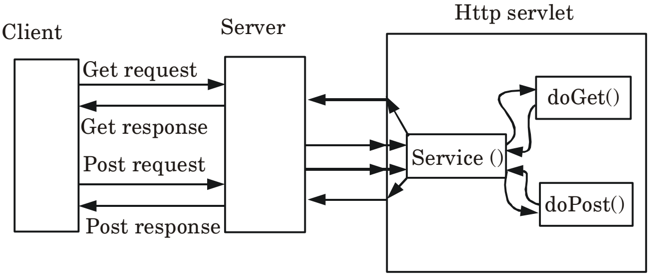


Fig. 5.7.1.

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

Que 5.8. **Explain servlets with its life cycle. How its life cycle is different from the life cycle of JSP ? Explain with an example.**

AKTU 2019-20, Marks 07

Answer

Servlets with its life cycle : Refer Q. 5.1, Page 5–2D, Unit-5.

Difference :

1. In servlet life cycle, the servlet object is created first.
2. The init() method is invoked by the servlet container and the servlet is initialized by its arguments.
3. Servlet's service() method is invoked next. At the end, the destroy() method is invoked.
4. In case of a Java Server Page life cycle, the .jsp is converted into .class file which is a servlet and then follows the process of the servlet. In other words, the .jsp is translated into servlet and the functionality is same as that of the servlet.

   JSP lifecycle example :

   Demo.jsp :

   ```
   <html>
   <head>
   <title>Demo JSP</title>
   </head>
   <%
   int demvar=0;%>
   <body>
   Count is :
   <% Out.println(demovar++); %>
   <body>
   </html>
   ```

Demo JSP Page is converted into demo_jsp.servlet in the below code.

```
Public class demp_jsp extends HttpServlet{
Public void_jspservice(HttpServletRequest request, HttpServletResponse
response) Throws IOException, ServletException
{
PrintWriter out = response.getWriter();
response.setContentType("text/html");
out.write ("<html><body>");
int demovar=0;
out.write("Count is:");
out.print(demovar++);
out.write("</body></html>");
}
}
```

5. The web container calls the init() method only once after creating the servlet instance.
6. The init() method is used to initialize the servlet. It is the life cycle method of the javax.servlet.Servlet interface.
7. Syntax of the init() method :

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

8. A new thread is then gets created, which invokes the_jspService() method, with a request (HttpServletRequest) and response (HttpServletResponse) objects as parameters shown below :

   ```
   [code language="java"]
   void_jspService( HttpServletRequest req, HttpServletResponse res)
   {
   //code goes here
   }[/code]
   ```

9. Invokes the jspDestroy() method to destroy the instance of the servlet class. code :

   ```
   [code language="java"]
   public void jspDestory()
   {
   //code to remove the instances of servlet class
   }
   ```

PART-2

Handling HTTP Get Requests, Handling Post Requests, Redirecting Requests to Other Resources.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 5.9. **Explain handling in HTTP get and post request in servlet with example.**

Answer

1. 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.
2. The methods that handle these requests are doGet() and doPost().

Handling Get requests :

1. Handling Get requests involves overriding the doGet() method.

   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();
}
...
}
```

2. The servlet extends the HttpServlet class and overrides the doGet() method.
3. Within the doGet() method, the getParameter () method gets the servlet's expected argument.

4. To respond to the client, the example doGet() method uses a Writer from the HttpServletResponse object to return text data to the client.
5. At the end of the doGet() method, after the response has been sent, the Writer is closed.

Handling Post requests :

1. Handling Post requests involves overriding the doPost() method.

   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>" +
...);
out.println("<h3>Thank you for purchasing your books from us" +
        request.getParameter("cardname") +
                ...);
out.close();
}
...
}
```

2. The servlet extends the HttpServlet class and overrides the doPost() method.
3. Within the doPost() method, the getParameter() method gets the servlet's expected argument.
4. To respond to the client, the example doPost() method uses a Writer from the HttpServletResponse object to return text data to the client.
5. At the end of the doPost() method, after the response has been set, the Writer is closed.

Que 5.10. Explain RequestDispatcher. Also describe different ways to get the object of RequestDispatcher.

Answer

1. The RequestDispatcher is a interface which provides the facility of dispatching the request to another resource such as HTML, servlet or JSP.
2. This interface can also be used to include the content of another resource also.
3. It is one of the ways of servlet collaboration.

Methods of RequestDispatcher interface :

1. **Public void forward(ServletRequest request, ServletResponse response)throws ServletException, java.io.IOException :** It forwards a request from a servlet to another resource (servlet, JSP file, or HTML file) on the server.
2. **Public void include(ServletRequest request, ServletResponse response)throws ServletException, java.io.IOException :** It includes the content of a resource (servlet, JSP page, or HTML file) in the response.

There are three ways to get RequestDispatcher object in the servlet :

1. **By calling the getRequestDispatcher(String path) method on ServletRequest :** It returns a RequestDispatcher object that wraps the resource located at the given path. This method will return null if the ServletContext cannot return a RequestDispatcher.
2. **By getRequestDispatcher(String path) method on ServletContext :** It returns a RequestDispatcher object that wraps the named servlet or JSP page. Names can be defined for servlets and JSP pages in the web application. If the ServletContext cannot return a RequestDispatcher, this method will return null.
3. **By calling getNamedDispatcher(servlet logical name) on ServletContext object :** It returns a RequestDispatcher object that acts as a wrapper for the resource located at the specified context-relative path.

PART-3

Session Tracking, Cookies, Session Tracking with HTTP Session.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 5.11. **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.
3. Each time user requests to the server, server treats the request as the new request and maintain the state of a 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. 5.11.1.

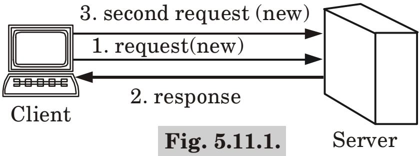


Fig. 5.11.1.

5. There are four techniques used in session tracking :
   a. Cookies
   b. Hidden Form Field
   c. URL Rewriting
   d. HttpSession

Que 5.12. Explain cookies with its advantages and disadvantages. Also give its type.

Answer

1. Cookie is a key value pair of information sent by the server to the browser which is saved in the client computer.
2. A cookie is a small piece of information that is persisted between the multiple client requests.
3. 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 :

1. Client sends a request to the server.
2. Cookie is added with request from the servlet and stored in the cache of the browser.
3. After that if response is sent by the server, cookie is added with it by default.
4. Server can identify the client using the cookie.

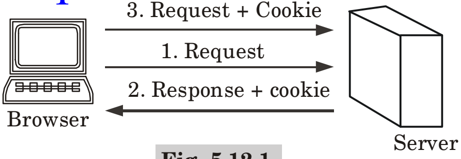


Fig. 5.12.1.

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 sign out.

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 5.13. **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.

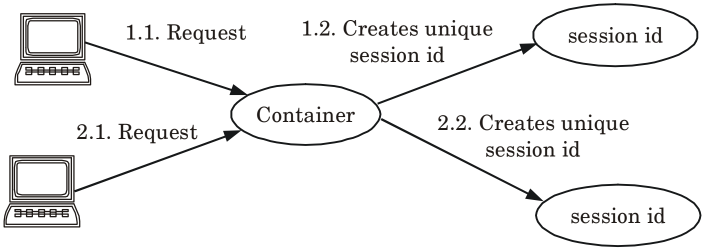

1.1. Request
1.2. Creates unique session id
session id
Container
2.1. Request
2.2. Creates unique session id
session id

Fig. 5.13.1.

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 5.14. **What is difference between session and cookies ? Write a servlet program for servlet login and logout using cookies.**

AKTU 2018-19, Marks 07

Answer

S. No.	Session	Cookies
1.	Sessions are stored in server.	Cookies are stored in the user's browser.
2.	A session is available as long as the browser is opened. User can not disable the session.	Cookies can keep information in the user's browser until deleted by user or set as per the timer.
3.	It will be destroyed if we close the browser.	It will not be destroyed even if we close the browser.
4.	It can store any object.	Cookies can only store string.
5.	Session cannot be same for future reference.	We can save cookies for future reference.

Program :

Index.html :

```
<!DOCTYPE html><html><head>
```

```
<meta charset="ISO-8859-1">
<title>Servlet Login Example</title>
</head><body>
<h1>Welcome to Login App by Cookie</h1>
<a href="login.html">Login</a>|
<a href="LogoutServlet">Logout</a>|
<a href="ProfileServlet">Profile</a></body></html>
```

link.html :

```
<a href="login.html">Login</a> |
<a href="LogoutServlet">Logout</a> |
<a href="ProfileServlet">Profile</a>
<hr>
```

login.html :

```
<form action="LoginServlet" method="post">
Name:<input type="text" name="name"><br>
Password:<input type="password" name="password"><br>
<input type="submit" value="login"></form>
```

LoginServlet.java :

```
package com.javatpoint;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class LoginServlet extends HttpServlet {
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out=response.getWriter();
request.getRequestDispatcher("link.html").include(request, response);
String name=request.getParameter("name");
String password=request.getParameter("password");
if(password.equals("admin123")){
```

```
out.print(“You are successfully logged in!”);
out.print(“<br>Welcome, ”+name);
Cookie ck=new Cookie(“name”,name);
response.addCookie(ck);
}else{
out.print(“sorry, username or password error!”);
request.getRequestDispatcher(“login.html”).include(request, response); }
out.close(); } }
```

LogoutServlet.java :

```
package com.javatpoint;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class LogoutServlet extends HttpServlet {
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
response.setContentType(“text/html”);
PrintWriter out=response.getWriter();
request.getRequestDispatcher(“link.html”).include(request, response);
Cookie ck=new Cookie(“name”,“”);
ck.setMaxAge(0);
response.addCookie(ck);
out.print(“you are successfully logged out!”); } }
```

Que 5.15. 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”); PreparedStatement
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>”);
}
ut.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-4

Java Server Page (JSP) : Introduction, Java Server Page Overview, A First Java Server Page Example.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 5.16. What do you mean by JSP ? Explain the architecture of JSP. How JSP provides better performance ?

Answer

1. 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.
2. JSP supports both scripting and element based dynamic content, and allows developers to create their own tag libraries.
3. JSP pages are compiled for efficient server processing.
4. JSP is platform independent. So, it can be easily upgraded or switched without affecting JSP based applications.

JSP architecture :

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

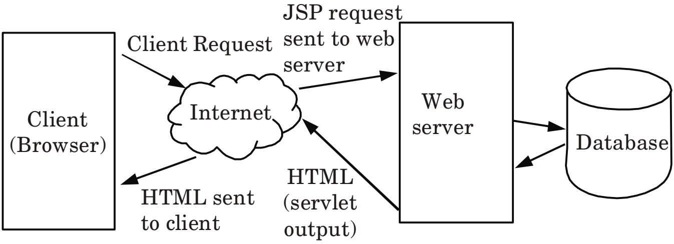


Fig. 5.16.1. JSP architecture.

a. The user goes to a JSP page and makes the request via internet in web browser.

b. The JSP request is sent to the web server.

c. Web server accepts the requested .jsp file and passes the JSP file to the JSP Servlet Engine.

d. 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.

d. 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 5.17. What is JSP ? What are the advantages of JSP over various server-side programs ?

Answer

JSP : Refer Q. 5.16, Page 5–20D, Unit-5.

Advantages of JSP over other server-side programs are :

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

Que 5.18. Explain JSP processing.

Answer

1. 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.
2. JSP pages can be processed using JSP container only.
3. Following are the steps that need to be followed while processing the request for JSP page :

   a. Client makes a request for required JSP page to the server.

   b. The server must have JSP container so that JSP request can be processed.

c. On receiving request the JSP container searches and then reads the desired JSP page.

d. Then JSP page is converted to corresponding servlet. Basically any JSP page is a combination of template text and JSP element.

e. Every template text is translated into corresponding println statement.

Que 5.19. **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 is a web server and servlet container that is used to deploy and serve Java web application.
2. Tomcat server is a Java-capable HTTP server, which could execute special Java programs known as "Java servlet" and "Java Server Pages".
3. Tomcat can operate as a standalone web server.
4. It can operate as an out-of-process servlet container for some web servers, such as Apache.
5. For other web servers, such as IIS (Internet Information Services), it can operate as an in-process servlet container.
6. Tomcat server runs on a specific TCP post from a specific IP address.
7. The default TCP port number for HTTP protocol is 80, which is used for the production HTTP server.
8. To test HTTP server, we can choose any unused port number between 1024 and 65535.
9. There are two important environment variables to set before running Tomcat :

   a. Set CATALINA_HOME to the root of Tomcat directory.

   b. Set JAVA_HOME to the root directory of Java JDK or JRE.

Que 5.20. **What are the steps for running JSP program in Tomcat server ?**

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 5.21. Write a simple JSP page for displaying the message : "This is my first JSP page !!!".

Answer

```
<%@ page language = "java" contentType = "test/html" %>
<%@ 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>
```

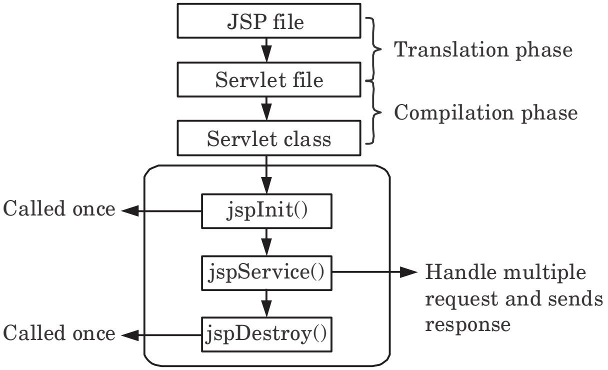
Que 5.22. Compare JSP and Servlet. Explain the life cycle of a JSP page with a suitable diagram. Also list any five action tags used in JSP.

AKTU 2018-19, Marks 07

Answer

S. No.	JSP	Servlet
1.	JSP is protocol dependent.	Servlet is protocol independent.
2.	It can handle only HTTP and HTTPS protocol.	It can handle any type of protocol *i.e.*, FTP, HTTP.
3.	Time taken to generate response for first request is more.	Time taken to generate response for first request is less.
4.	JSP is a scripting language which can generate dynamic response.	Servlet is a Java program which can generate dynamic response.
5.	It is easy to code.	It is not easy to code.

Life cycle of JSP :



1. **Translation of JSP page to Servlet :** This is the first step of JSP life cycle. This translation phase deals with syntactic correctness of JSP. Here test.jsp file is translated to test.java.
2. **Compilation of JSP page :** Here the generated java servlet file (test.java) is compiled to a class file (test.class).
3. **Class loading :** Servlet class which has been loaded from JSP source is now loaded into container.
4. **Instantiation :** Here instance of the class is generated. The container manages one or more instance by providing response to requests.
5. **Initialization :** jspInit() method is called only once during the life cycle immediately after the generation of servlet instance from JSP.
6. **Request processing :** jspService() method is used to serve the raised requests by JSP. It takes request and response object as parameters. This method cannot be overridden.
7. **JSP cleanup :** In order to remove the JSP from use by the container or to destroy method for servlets jspDestroy() method is used. This method is called once, if we need to perform any cleanup task like closing open files, releasing database connections jspDestroy() can be overridden.

Five action tags used in JSP are :

1. <JSP : use Bean>
2. <JSP : set Property>
3. <JSP : include>
4. <JSP : attribute>
5. <JSP : body>

Que 5.23. How we deploy the Servlets on Tomcat Web Container ? Also explain how we change the default number of Tomcat Container ?

Answer

To create a Servlet applications do the following steps :

1. Create directory structure for our application.
2. Create a Servlet.
3. Compile the Servlet.
4. Create deployement descriptor for our application.
5. Start the server and deploy the application.

Following steps are required to change the default port of Tomcat container :

1. Stop Tomcat server if it is already running.
2. Open <Tomcat_Home>/conf/server.xml file for editing.
3. Look for port *8080* in the xml file and replace with any available port.
4. If there is other Tomcat running on the same server, we have to change SHUTDOWN port as well.
5. Start Tomcat server.

Que 5.24. **JSP is an extension of Servlets not replacement, Justify ? What problems of Servlets technology can JSP is suppose to solve ?**

Answer

JSP is an extension of servlet not replacement. This statement can be justified by following advantages of JSP :

1. JSP is not a replacement of Servlets but extension of Servlets, as coding decreases more than half.
2. In JSP, static code and dynamic code are separated.
3. JSP needs no compilation by the programmer.
4. In JSP, presentation layer and business logic layer can be separated with the usage of Java Beans.

Problem of Servlet technology solved by JSP are :

1. Difficult to code.
2. Do not allow parsing and decoding HTML headers.
3. It cannot be integrated with other backend services.
4. It does not manage cookies.
5. Do not allow reading and sending HTML headers.

PART-5

Implicit Objects, Scripting, Standard Actions, Directives, Custom Tag Libraries.

Questions-Answers

Long Answer Type and Medium Answer Type Questions

Que 5.25. **Explain implicit objects available in JSP with example.**

AKTU 2017-18, Marks 10

Answer

Different types of implicit objects in JSP are :

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">
```

```
<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>:<%=config.getInitParameter(“User”) %><br><br>
<%-- application object example --%>
<strong>User context param value</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() %>
</body>
</html>
```

Que 5.26. **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

OR

What are standard actions in JSP ? Illustrate with example.

AKTU 2017-18, Marks 10

Answer

Java Beans and why they are used : Refer Q. 4.4, Page 4–5D, Unit-4.

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.
6. **<jsp:body> :**
   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/ gets 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>
<head><title>jsp:text action</title></head>
<body>
```

```
<books><book><jsp:text>
Welcome to JSP Programming
</jsp:text></book></books>
</body>
</html>
```

Que 5.27. **What do you mean by JSP processing ? How JSP pages are handled ? Explain various JSP directives with suitable examples.**

AKTU 2015-16, Marks 10

OR

What are JSP directives ? Explain various types of directives with example.

AKTU 2017-18, Marks 10

Answer

JSP processing : Refer Q. 5.18, Page 5–21D, Unit-5.

JSP page handling :

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" %>

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:

   ```
   <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 5.28. **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%>`

      iv. It is noticed that there is no space between <% and =.

   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 5.29. **Explain the difference between JSP includes directive elements and JSP includes action elements.**

Answer

S. No.	JSP include directive element	JSP include action element
1.	Resources are included at translation time.	Resources are included at request time.
2.	It uses file attribute to specify the resource.	It uses page attribute to specify the resource.
3.	Used to include static resource such as HTML.	Used to include dynamic resource such as JSP.
4.	It does not allow to pass parameters.	It allow to pass parameter using <jsp:param> action tag.
5.	It does not pass request and response object.	It allows us to pass request and response object.
6.	It can use both absolute and relative path to include resource.	It can use only relative path to include resource.

Que 5.30. **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. 4.4, Page 4–5D, 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 :

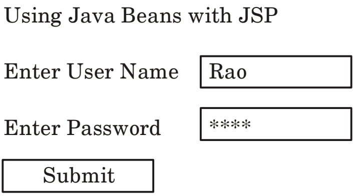
```
<body>
<jsp:useBean id="snr" class="pack.ValidateBean" />
<jsp:setProperty name="snr" property="user" />
<jsp:setProperty name="snr" property="pass" />
You entered user name as <jsp:getProperty name="snr" property="user" />
<br>
You entered user password as <jsp:getProperty name="snr" property="pass"
/> <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: Rao

Enter Password: ****

```
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 the 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 the output stream of client.

   <%= snr.validate(“Rao”, “java”) %>
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 5.31. **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.

<mytaglib:newtag>

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 5.32. **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.

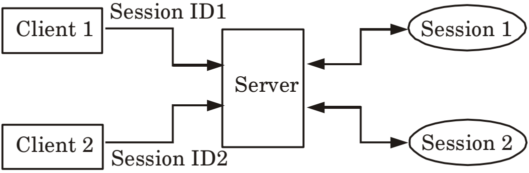


Fig. 5.32.1. Session scope.

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.

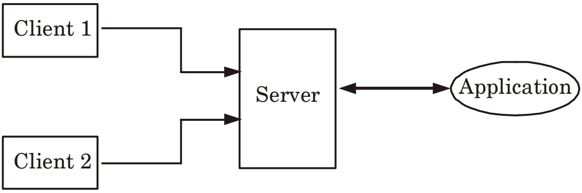


Fig. 5.32.2. Application scope.

VERY IMPORTANT QUESTIONS

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

Q. 1. Explain the life cycle of servlet. Also write a servlet for displaying a string "HELLO WORLD!"

Ans. Refer Q. 5.2.

Q. 2. What is servlet ? Explain its life cycle. Illustrate some characteristics of servlet. How does servlet score over CGI ?

Ans. Refer Q. 5.4.

Q. 3. Explain how servlet works.

Ans. Refer Q. 5.7.

Q. 4. What is difference between session and cookies ? Write a servlet program for servlet login and logout using cookies.

Ans. Refer Q. 5.14.

Q. 5. Write note on Tomcat server.

Ans. Refer Q. 5.19.

Q. 6. Compare JSP and Servlet. Explain the life cycle of a JSP page with a suitable diagram. Also list any five action tags used in JSP.

Ans. Refer Q. 5.22.

Q. 7. Explain implicit objects available in JSP with example.

Ans. Refer Q. 5.25.

Q. 8. What are the different types of standard action tags used in JSP ?

Ans. Refer Q. 5.26.

Q. 9. What do you mean by JSP processing ? How JSP pages are handled ? Explain various JSP directives with suitable examples.

Ans. Refer Q. 5.27.

☺☺☺

1 UNIT

Introduction and Core Java (2 Marks Questions)

1. 1. What is internet ? Discuss the various internet services in brief.

AKTU 2015-16, Marks 02

Ans. Internet is a global system of interconnected computer networks that use the standard Internet Protocols Suite (TCP/IP) to serve billions of users worldwide.

Various internet services are :

1. Communication services
2. Information retrieval services
3. Web services
4. World Wide Web

1.2. What do you mean by web projects ? Describe the various protocol governing web projects.

AKTU 2015-16, Marks 02

OR

What is web project ?

AKTU 2017-18, Marks 02

Ans. A web project is the process of developing and creating different types of websites that implemented on the internet.

Protocols governing web project :

1. FTP
2. HTTP
3. Telnet
4. POP3
5. SMTP
6. SNMP

1.3. Explain the client-server architecture with diagram.

AKTU 2015-16, 2017-18; Marks 02

Ans. In client-server architecture, the clients are programs running on remote machines that communicate with a program called the server that runs on a single site and responds to requests from many clients.

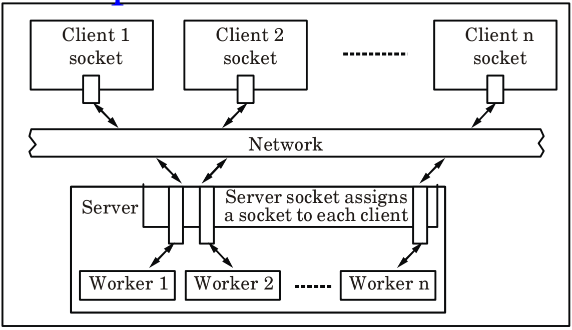


Fig. 1.3.1.

1.4. How do you call a package ? **AKTU 2016-17, Marks 02**

Ans. There are two ways to call the package :

1. Import packagename; 2. Import package.classname;

1.5. Give some name of object-oriented programming languages.

Ans. Object-oriented programming languages are :

1. C++
2. Java
3. Simula
4. SmallTalk
5. Ada

1.6. What is an exception ? How the exception is handled in Java ? Differentiate between throw and throws.

AKTU 2015-16, Marks 02

Ans. An exception is an event, which occurs during the execution of a program that disrupts the normal flow of program instruction.

Exception is handled in Java by using following five keywords :

1. try
2. catch
3. finally
4. throw
5. throws

Difference between throw and throws :

S. No.	Throw	Throws
1.	Throw keyword is used to throw an exception explicitly.	Throws clause is used to declare an exception.

2.	General form of throw is :
	Throw throwable instance;
		General form of throws is :
		type method_name(parameter-list) throws exception-list
		{
		// body of method
		}

1.7. What is an applet ? How do applets differ from application programs ? Discuss the life cycle of applet.

AKTU 2015-16, Marks 02

Ans. Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Difference between applets and application programs :

S. No.	Applets	Application program
1.	An applet is a Java program which is embedded in web page to generate dynamic content.	An application is a Java program that runs independently on client/server without web browser.
2.	The execution of the program does not start from main ( ) method.	The execution of the program starts from the main( ) method.

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

1. init( )
2. start( )
3. stop( )
4. destroy( )
5. paint( )

1.8. What is thread ? How to create a thread in Java ?

AKTU 2015-16, Marks 02

Ans. A thread is a light weighted process which runs concurrently with other threads. All threads of program define a separate path of execution.

Create a thread in Java :

```
public class Thread Test
{
public static void main (string [ ] args)
{
System.out.println (“Constructing the thread....”);
BytePrinter bp = new BytePrinter ( );
System.out.println (“Starting the thread....”);
bp start ( );
System.out.println (“The thread has been started”);
System.out.println (“The main ( ) method is finishing”);
return;
}
}
```

1.9. What do you mean by AWT ? Explain the working of AWT.

AKTU 2015-16, Marks 02

Ans. AWT is a library of classes which provides GUI tools to develop GUI applications and applets.

Working of AWT : Working of AWT can be explained with the help of program to create a window in Java application :

```
import java.awt.*;
class AWTexample extends Frame
{
public AWTexample ( )
{
setTitle ("My window");
setSize (300, 200);
}
public static void main (string [ ] args)
{
AWTexample F = new AWTexample ( );
F. setVisible (true);
}
}
```

1.10. What is the difference between JDK, JRE, JIT and JVM ?

AKTU 2015-16, Marks 02

Ans.

1. Java Development Kit (JDK) is a bundle of software components that is used to develop Java based applications.
2. Java Runtime Environment (JRE) is an implementation of the JVM which actually executes Java programs.
3. Just In Time compiler (JIT) compiler is a component of JRE which improves the performance of Java applications by compiling byte code to native machine code at runtime.
4. Java Virtual Machine (JVM) is an abstract computing machine.

1.11. What is bytecode ? **AKTU 2018-19, Marks 02**

Ans. Bytecode is a program code that has been compiled from source code into low-level code and designed for a software interpreter.

1.12. Define constructor. **AKTU 2018-19, Marks 02**

Ans. A constructor is a special type of subroutine called to create an object. It prepares the new object for use, often accepting arguments that the constructor uses to set required member variables.

1.13. What is an instance variable ? **AKTU 2018-19, Marks 02**

Ans. An instance variable is a variable defined in a class, for which each instantiated object of the class has a separate copy, or instance. An instance variable is similar to a class variable.

1.14. What are the two major protocols for accessing e-mail from servers ?

AKTU 2018-19, Marks 02

Ans. Post Office Protocol version 3 (POP3), Simple Mail Transfer Protocol (SMTP) are the two major protocols used for accessing e-mail from servers.

1.15. What do you mean by checked exceptions ?

AKTU 2018-19, Marks 02

Ans. A checked exception is a type of exception that must be either caught or declared in the method in which it is thrown. For example, the java.io.IOException is a checked exception.

1.16. Packages and interfaces both acts as a container. How ?

Ans. Container is a component which contains other components inside itself. So packages contain a group of related classes and an interface contains methods and fields. The content in packages and interfaces can be used by the classes by importing it in our program. This is how packages and interfaces both act as a container.

1.17. Define Path in JDK. How path is different from class path ?

AKTU 2019-20, Marks 02

Ans. Path variable is set for providing path for all Java tools like Java, Javac, Javap, Javah, jar, Appletviewer which are used in Java programming. All these tools are available in bin folders so we set path upto bin folders.

S. No.	Path	Class path
1.	Path variable is set for providing path for all java tools like Java, Javac, Javap, Javah, jar, Appletviewer	Class path variable is set for providing path of all Java classes which is used in our application.
2.	It contains a path to the Java tools.	It contains a path of the classes provided by JDK.
3.	Java tools include Java, Javac, Javap, Javah.	All the classes are available in "rt.jar" file.
4.	Command to check path : >>echo %path%.	Command to check class path : >>echo %classpath%

1.18. Describe garbage collection and demonstrate how it is functioning ? **AKTU 2019-20, Marks 02**

Ans. Garbage Collection (GC) is a form of automatic memory management. The garbage collector attempts to reclaim garbage, or memory occupied by objects that are no longer in use by the program.

Functioning of garbage collection :

1. In the first step, unreferenced objects are identified and marked as ready for garbage collection.
2. In the second step, marked objects are deleted.
3. Memory can be compacted after the garbage collector deletes objects, so remaining objects are in a contiguous block at the start of the heap.
4. The compaction process makes it easier to allocate memory to new objects sequentially after the block of memory allocated to existing objects.

☺☺☺

2

UNIT

Web Page Designing (2 Marks Questions)

2.1. Explain the HTML tags : Table, frame and forms with the help of suitable examples. **AKTU 2015-16, Marks 02**

Ans. **Table :**

A table is a two dimensional matrix, consisting of rows and columns.

Example :

```
<table><tr>
<td> row 1</td>
<td> row 1</td>
</tr>
</table>
```

Frame :

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.

Example :

```
<framset cols = "50%, 50%">
<frame src = "File 1.html">
<frame src = "File 2.html">
</framset>
```

Forms :

All the input elements should be enclosed within the opening and closing tags like this :

<FORM> The input elements go here </FORM>

Example :

```
<form>
Login :
<input type = "text" name = "login" size = " 30">
Password :
<input type = "password" name = "password" >
<input type = "submit" value = "Submit">
</form>
```

2.2. What is CSS ? What are different ways of creating stylesheets ? **AKTU 2015-16, Marks 02**

Ans. CSS is a language that describes the style of the HTML document. Different ways of creating stylesheets are :

1. Inline stylesheets
2. Embedded stylesheets
3. External stylesheets

2.3. What is HTML ? **AKTU 2016-17, Marks 02**

Ans. HTML (Hypertext Markup Language) is a set of special codes that can be embedded in text to add formatting and linking information.

2.4. What is selector ? Mention its types.

AKTU 2016-17, Marks 02

Ans. A selector is a string that identifies what elements the corresponding rule is applied.

Types of selectors :

1. Simple selectors
2. Combinator selectors
3. Pseudo-class selectors
4. Pseudo-elements selectors
5. Attribute selectors

2.5. What is the usage of CSS ? **AKTU 2016-17, Marks 02**

Ans. CSS is used to enhance the features of web pages like formatting styles and thereby increase the visual appearance of web pages. CSS helps in faster downloading of the pages.

2.6. What are forms and how they are created in HTML ?

AKTU 2017-18, Marks 02

Ans. An HTML form is a section of document containing normal content, markup, special elements called controls (checkboxes, radio buttons, menus etc.) and labels on those controls.

The HTML <form> tag is used to create an HTML form and it has following syntax :

```
<form action = "script URL" method = "GET | POST">
Form elements like input, text area etc.
</form>
```

2.7. Explain the anchor and table tag in HTML.

AKTU 2017-18, Marks 02

Ans. **Anchor tag :**

1. The HTML anchor tag defines a hyperlink that links one page to another page.
2. The "href" attribute is used to specify a target for the anchor tag.

3. The syntax of HTML anchor tag :

<a href='......'> Link Text</a>

Table tag : Refer Q. 2.1, Page SQ–7Z, Unit-2, Two Marks Questions.

2.8. Explain well formed and valid XML.

AKTU 2017-18, Marks 02

Ans. An XML document with correct syntax is called well formed XML. An XML document is valid if it is validated against a DTD.

2.9. Explain the difference between id and class selector in CSS.

AKTU 2017-18, Marks 02

Ans.

S. No.	Id selector	Class selector
1.	It is used to specify a style for a unique element.	It is used to specify a style for a group of element.
2.	The syntax for id selector is : #idselector_name {property:value;}	The syntax of class selector is : class_selector_name {property:value;}

2.10. Differentiate between dynamic web pages and static web pages.

AKTU 2016-17, Marks 02

Ans.

S. No.	Dynamic web page	Static web page
1.	Dynamic web pages are behavioral and have the capacity to produce distinctive content for different visitors.	Static web pages will remain same for the time until and unless someone changes it manually.
2.	Dynamic pages are complex to construct.	Static web pages are simple to construct.

2.11. When is it appropriate to use frames ?

AKTU 2018-19, Marks 02

Ans. When user wants to view multiple documents within a single web page, then it is appropriate to use frames.

2.12. What is the use of alternative text in image mapping ?

AKTU 2018-19, Marks 02

Ans. The alternative text in image mapping is use to provides alternative information for an image if image cannot be displayed.

2.13. Explain the difference between portal and website.

Ans.

S. No.	Portal	Website
1.	A specially designed website that brings information from diverse sources like emails, online forums and search engines together in a uniform manner.	A collection of related web pages, including multimedia content, typically identified with a common domain name and published on at least one web server.
2.	Provides essential services such as search engine capabilities, exchanging e-mails, links to other related sites and personalized content.	Allow the user to log in, register, obtain services according to the website type, sending feedbacks, etc.

2.14. What is the difference between <div> and <span> tag in HTML ?

Ans.

S. No.	<div> tag	<span> tag
1.	It is a block level element.	It is an inline element.
2.	It is a generic container for a block of HTML.	It is a generic container for a text or inline HTML.

2.15. We want to submit the data on page itself, what can be done ?

Ans. We can use GET() and POST() method to submit the data on page itself.

2.16. What is the role of CGIs in web design ?

Ans. Role of Common Gateway Interface (CGI) in web design :

i. It helps the web server to pass the user request to an application program.

ii. It helps to receive data from the server and forward to the user.

2.17. Define DHTML with suitable example.

AKTU 2019-20, Marks 02

Ans. DHTML : DHTML is a collection of technologies used together to create interactive and dynamic websites.

Example :

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<title>Welcome to Quantum Page </title>
</head>
<body>
<h1 onclick="this.style.color='red'">Hi Everyone!</h1>
</body>
</html>
```

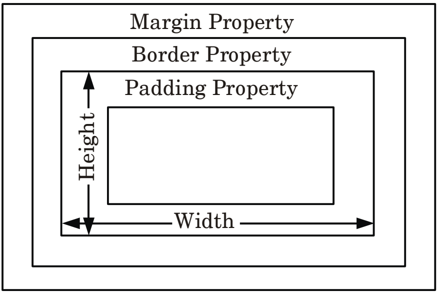
2.18. Define box model in CSS with block diagram.

AKTU 2019-20, Marks 02

Ans. CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of webpages. It can be used as a toolkit for customizing the layout of different elements.

Block diagram :

CSS Box-Model Property



2.19. Define webpage with its type. Discuss responsive webpage with example.

AKTU 2019-20, Marks 02

Ans. A webpage is a document commonly written in HTML that is accessible through the Internet or other networks using an Internet browser.

Types of webpage :

1. **Dynamic webpages :** Dynamic webpages are behavioral and have the capacity to produce distinctive content for different visitors.
2. **Static webpages :** Static webpages will remain same for the time until and unless someone changes it manually.

Responsive webpage :

1. A responsive webpage is one that has been designed to respond, or adapt, based on the technology and type of computing device used by the visitor to display the site.

2. It is basically one website design that will look good at any size from a large desktop LCD monitor to the smaller screens we use on smart phones and tablets.
3. Facebook, Youtube are the example of responsive webpage.

2.20. How XML is different from HTML ?

Ans.

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.

☺☺☺

3 UNIT

Scripting and Networking (2 Marks Questions)

3.1. Give any three uses of JavaScript.

Ans. Three uses of JavaScript are :

1. JavaScript can be used to create some web applications such as calculator, calendar, paint like application.
2. It can be used to detect the visitor's browser.
3. It can be used to validate the data.

3.2. How you will define the term function in JavaScript ?

Ans.

1. We can define the function anywhere in the script either in head or body section or in both.
2. The keyword function is used while defining the function.
3. The syntax for defining the function is :

```
function function_name (arg1, arg2,...argn)
{
. . .
Statements
}
```

3.3. Are Java and JavaScript the same ?

Ans. No, Java is an Object-Oriented Programming (OOP) language while JavaScript is a scripting language. Java creates applications that run in a virtual machine or browser while JavaScript code runs on a browser only.

3.4. What do you understand by AJAX ?

Ans. AJAX (Asynchronous JavaScript And XML) is used for allowing the client-side application to communicate with the server-side application.

3.5. Mention some predefined objects in JavaScript.

Ans. Some predefined objects in JavaScript are :

a. Number
b. String
c. Boolean

3.7. What is internet address ?

Ans. An internet address is a number that uniquely identifies each computer on the network.

3.7. Mention some classes that are present in java.net package.

Ans. Some classes present in java.net package are :

1. DatagramPacket
2. InetAddress
3. ServerSocket
4. SocketAddress
5. URLClassLoader
6. Inet4Address
7. JarURLConnection

3.8. Define URLConnection class.

Ans. URLConnection is a general purpose class for accessing the attributes of a remote resource.

3.9. Mention some methods for accessing DatagramPacket object.

Ans. Some methods for accessing DatagramPacket object are :

a. InetAddress getAddress ( )

b. int getPort ( )

c. int getLength ( )

3.10. How a JavaScript works ? **AKTU 2016-17, Marks 02**

Ans.

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

3.11. Define factory method and instance method ?

Ans. Factory method deals with the problem of creating objects without specifying the exact class of the object that will be created.

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

3.12. Differentiate the terms SendRequest() from Hyperlink.

Ans. SendRequest() : It is a method used in JavaScript. It is used to send HTTP request to the remote server.

Hyperlink : It is an icon, graphic or text in a document that links to another file or object.

☺☺☺

4 UNIT

Enterprise Jave Beans and JDBC (2 Marks Questions)

4.1. What is EJB ? Explain its types. **AKTU 2017-18, Marks 02**

Ans. Enterprise Java Bean (EJB) is the server-side and platform independent Java application programming interface (API) for Java Platform Enterprise Edition (Java EE). EJB is used to simplify the development of large distributed applications.

Types of EJB are :

1. Entity Bean
2. Session Bean
3. Message driven Bean

4.2. What do you mean by Java Bean ? Give the advantage of Java Bean.

Ans. Java Bean is the programming components written in Java. These components can be embedded in the scripting language like JSP.

Advantage of Java Bean :

1. A bean obtain all the benefits of Java's "write once, run anywhere" paradigm.
2. The properties, events and methods of a bean that are exposed to application builder tool can be controlled.

4.3. What are the properties of Java Bean ?

Ans. Properties of Java Bean are :

a. **Simple properties :** A simple property has a single value.

b. **Boolean properties :** A boolean property has a value of true or false.

c. **Indexed properties :** An indexed property consist of multiple value.

4.4. Differentiate between Java Bean and EJB.

Ans.

S. No.	Java Bean	EJB
1.	The Java Bean components can be visible or invisible to the user.	The EJB components are always invisible to the user.
2.	The Java Bean normally runs locally.	EJB are distributed components and always exist on the servers.

4.5. Define JDBC.

Ans. JDBC (Java Database Connectivity) is an API (Application Programming Interface) which is used for connectivity of Java application with RDBMS packages.

4.6. Define ODBC. **AKTU 2016-17, Marks 02**

Ans. Open Database Connectivity (ODBC) is an open standard API for accessing a database. By using ODBC statements in a program, we can access files in a number of different databases, including MS Access, DB2, Excel, and Text.

4.7. What are JAR files ? Explain its advantage. Write the command for creating JAR file. **AKTU 2017-18, Marks 02**

Ans. A JAR (Java ARchive) file is a file that contains compressed version of class files, audio files, image files or directories.

Advantages of JAR file :

a. Security

b. Decreased download time

c. Compression

To create a JAR file, we can use following command :

jar cf jarfilename inputfiles

4.8. List some features of JAR file.

Ans. **Features of JAR files are :**

1. JAR files are used for installing software.
2. It can be compressed.
3. Each element of JAR file can be verified.

4.9. Write various types of drivers available in Java for database handling. **AKTU 2017-18, Marks 02**

Ans.

1. JDBC-ODBC bridge driver
2. Native API driver

3. Network protocol driver
4. JDBC Net driver

4.10. What are the types of session bean ?

Ans. Types of session bean are :

i. Stateless session bean
ii. Stateful session bean
iii. Singleton session bean

4.11. Compare JDBC and ODBC. List the different types of JDBC drivers.

AKTU 2019-20, Marks 02

Ans. Comparison :

S. No.	JDBC	ODBC
1.	JDBC is language and platform dependent.	ODBC is language and platform independent.
2.	JDBC is Java Database Connectivity	ODBC is Open Database Connectivity.
3.	Code is easy to understand.	Code is complex.

Different JDBC drivers : Refer Q. 4.9, Page SQ–16D, Unit-4, Two Marks Questions.

☺☺☺

5

UNIT

Servlets and Java Server Pages (JSP) (2 Marks Questions)

5.1. What do you mean by servlets ? What are the methods used in life cycle of servlets ?

Ans. Servlets are server-side components that provide a powerful mechanism for developing server-side of web application.

The life cycle of a servlet uses following three methods :

a. Init() method

b. Service() method

c. Destroy() method

5.2. Compare servlet with CGI. **AKTU 2016-17, Marks 02**

Ans.

S. No.	Servlet	CGI
1.	It is written in Java.	It is written in C, C++, Visual Basic.
2.	It is thread based.	It is process based.
3.	It is platform dependent.	It is platform independent.
4.	It delivers non-scalable application.	It delivers scalable application.

5.3. Illustrate some characteristics of servlets.

Ans.

1. Servlets are the programs at the server side.
2. The servlets are intended to response the applets or HTML program.
3. Servlets have no graphical user interface.
4. Servlets can be used in cookies and session tracking.

5.4. What is Java Server Pages technology ?

AKTU 2016-17, Marks 02

Ans. Java Server Pages (JSPs) are simple and powerful technology used to generate dynamic HTML documents on the server-side.

5.5. Write down the different JSP actions.

Ans. Different JSP actions are :

i. <jsp : param>

ii. <jsp : useBean>

iii. <jsp : include>

iv. <jsp : setProperty>

v. <jsp : forward>

5.6. Where the session information can be maintained in JSP ?

Ans. There are three methods using which the session information can be maintained :

1. By use of cookies.
2. By embedded hidden fields in a HTML form.
3. By sending URL string in response body.

5.7. Define Tomcat server. Write its component.

Ans. Tomcat server is an application server that executes Java servlets and renders web pages that include Java Server Page coding. Components of Tomcat server are :

1. Catalina
2. Coyote
3. Jasper

5.8. How to create cookie ?

Ans. Cookie is created by using following code :

```
cookie ck = new Cookie (“user”, “Aditya”) ;
response.addcookie(ck) ;
```

5.9. Mention some of the JSP implicit objects.

Ans. Some JSP implicit objects are :

1. Out object
2. Request object
3. Session object
4. Page object
5. Exception object

5.10. Discuss about Tomcat server. How to set the Class path for servlet in Tomcat server ? **AKTU 2019-20, Marks 02**

Ans. Tomcat server : Refer Q. 5.7, Page SQ–19D, Unit-5, Two Marks Questions.

To set class path for servlet in Tomcat server :

1. Copy servlet-api.jar file location and set the class path in environment variable.

2. If our class path is already set for core java programming, we need to edit class path variable. For edit classpath just put ‘;’ at end of previous variable and paste new copied location (without deleting previous classpath variable).

☺☺☺

B. Tech.

(SEM. V) ODD SEMESTER THEORY EXAMINATION, 2015-16

WEB TECHNOLOGY

Time : 3 Hours **Max. Marks : 100**

SECTION-A

Note : Attempt **all** parts. **All** parts carry equal marks. Write answer of each part in short. **(2 × 10 = 20)**

1. a. What is internet ? Discuss the various internet services in brief.

b. What do you mean by web projects ? Describe the various protocol governing web projects.

c. Explain the client-server architecture with diagram.

d. Explain the HTML tags : Table, frame and forms with the help of suitable examples.

e. What is CSS ? What are different ways of creating stylesheets ?

f. What is the difference between JDK, JRE, JIT and JVM ?

g. What is an exception ? How the exception is handled in Java ? Differentiate between throw and throws.

h. What is an applet ? How do applets differ from application programs ? Discuss the life cycle of applet.

i. What is thread ? How to create a thread in Java ?

j. What do you mean by AWT ? Explain the working of AWT.

SECTION – B

Note : Attempt any **five** questions from this section. **(10 × 5 = 50)**

2. What is XML ? Discuss the significance of XML. How is XML different from HTML ? Explain the process of XML parsing. How are they useful ?

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

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.**

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

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

7. **What do you mean by JSP processing ? How JSP pages are handled ? Explain various JSP directives with suitable examples.**

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

9. **Describe the PHP life cycle and how does PHP score over ASP and JSP ?**

SECTION – C

Note : Attempt any **two** questions from this section.

(15 × 2 = 30)

10. **Explain JDBC application architecture. What are the various types of JDBC drivers ? Write steps to connect database with the web application using JDBC.**

11. **What is PHP ? Describe PHP session in detail. Explain the steps involved in connecting database in PHP.**

12. **Write down the short note on the following :**

a. **COM/DCOM**

b. **Tomcat server**

c. **CORBA**

☺☺☺

SOLUTION OF PAPER (2015-16)

SECTION-A

Note : Attempt **all** parts. **All** parts carry equal marks. Write answer of each part in short. **(2 × 10 = 20)**

1. a. What is internet ? Discuss the various internet services in brief.

Ans. Internet is a global system of interconnected computer networks that use the standard Internet Protocols Suite (TCP/IP) to serve billions of users worldwide.

Various internet services are :

1. Communication services
2. Information retrieval services
3. Web services
4. World Wide Web

b. What do you mean by web projects ? Describe the various protocol governing web projects.

Ans. A web project is the process of developing and creating different types of websites that implemented on the internet.

Protocols governing web project :

1. FTP
2. HTTP
3. Telnet
4. POP3
5. SMTP
6. SNMP

c. Explain the client-server architecture with diagram.

Ans. In client-server architecture, the clients are programs running on remote machines that communicate with a program called the server that runs on a single site and responds to requests from many clients.

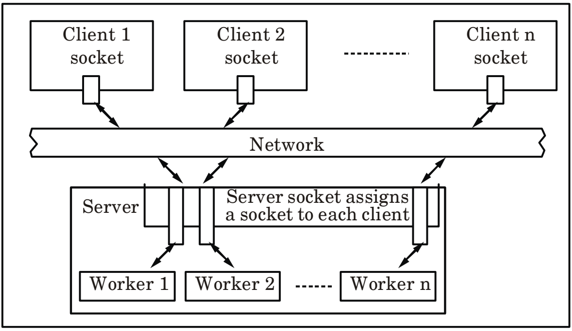


Fig. 1.

d. Explain the HTML tags : Table, frame and forms with the help of suitable examples.

Ans. **Table :**

A table is a two dimensional matrix, consisting of rows and columns.

Example :

```
<table><tr>
<td> row 1</td>
<td> row 1</td>
</tr>
</table>
```

Frame :

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.

Example :

```
<framset cols = “50%, 50%”>
<frame src = “File 1.html”>
<frame src = “File 2.html”>
</framset>
```

Forms :

All the input elements should be enclosed within the opening and closing tags like this :

<FORM> The input elements go here </FORM>

Example :

```
<form>
Login :
<input type = “text” name = “login” size = ” 30”>
Password :
<input type = “password” name = “password” >
<input type = “submit” value = “Submit”>
</form>
```

e. What is CSS ? What are different ways of creating stylesheets ?

Ans. CSS is a language that describes the style of the HTML document.

Different ways of creating stylesheets are :

1. Inline stylesheets
2. Embedded stylesheets
3. External stylesheets

f. What is the difference between JDK, JRE, JIT and JVM ?

Ans.

1. Java Development Kit (JDK) is a bundle of software components that is used to develop Java based applications.
2. Java Runtime Environment (JRE) is an implementation of the JVM which actually executes Java programs.

3. Just In Time compiler (JIT) compiler is a component of JRE which improves the performance of Java applications by compiling byte code to native machine code at runtime.
4. Java Virtual Machine (JVM) is an abstract computing machine.

g. What is an exception ? How the exception is handled in Java ? Differentiate between throw and throws.

Ans. An exception is an event, which occurs during the execution of a program that disrupts the normal flow of program instruction.

Exception is handled in Java by using following five keywords :

1. try
2. catch
3. finally
4. throw
5. throws

Difference between throw and throws :

S. No.	Throw	Throws
1.	Throw keyword is used to throw an exception explicitly.	Throws clause is used to declare an exception.

2.	General form of throw is :
	Throw throwable instance;
		General form of throws is :
		type method_name(parameter-list) throws exception-list
		{
		// body of method
		}

h. What is an applet ? How do applets differ from application programs ? Discuss the life cycle of applet.

Ans. Applet is a special type of program that is embedded in the webpage to generate the dynamic content. It runs inside the browser and works at client side.

Difference between applets and application programs :

S. No.	Applets	Application program
1.	An applet is a Java program which is embedded in web page to generate dynamic content.	An application is a Java program that runs independently on client/server without web browser.
2.	The execution of the program does not start from main ( ) method.	The execution of the program starts from the main( ) method.

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

1. init( )
2. start( )
3. stop( )

4. destroy( )
5. paint( )

i. What is thread ? How to create a thread in Java ?

Ans. A thread is a light weighted process which runs concurrently with other threads. All threads of program define a separate path of execution.

Create a thread in Java :

```
public class Thread Test
{
public static void main (string [ ] args)
{
System.out.println (“Constructing the thread....”);
BytePrinter bp = new BytePrinter ( );
System.out.println (“Starting the thread....”);
bp start ( );
System.out.println (“The thread has been started”);
System.out.println (“The main ( ) method is finishing”);
return;
}
}
```

j. What do you mean by AWT ? Explain the working of AWT.

Ans. AWT is a library of classes which provides GUI tools to develop GUI applications and applets.

Working of AWT : Working of AWT can be explained with the help of program to create a window in Java application :

```
import java.awt.*;
class AWTexample extends Frame
{
public AWTexample ( )
{
setTitle (“My window”);
setSize (300, 200);
}
public static void main (string [ ] args)
{
AWTexample F = new AWTexample ( );
F. setVisible (true);
}
}
```

SECTION – B

Note : Attempt any **five** questions from this section. **(10 × 5 = 50)**

2. What is XML ? Discuss the significance of XML. How is XML different from HTML ? Explain the process of XML parsing. How are they useful ?

Ans.

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>
<price>29.99</price>
</book>
<book category="web" cover="paperback">
<title lang="en">Learning XML</title>
```

```
<author>Erik T. Ray</author>
<year>2013</year>
<price>39.95</price>
</book>
</bookstore>
```

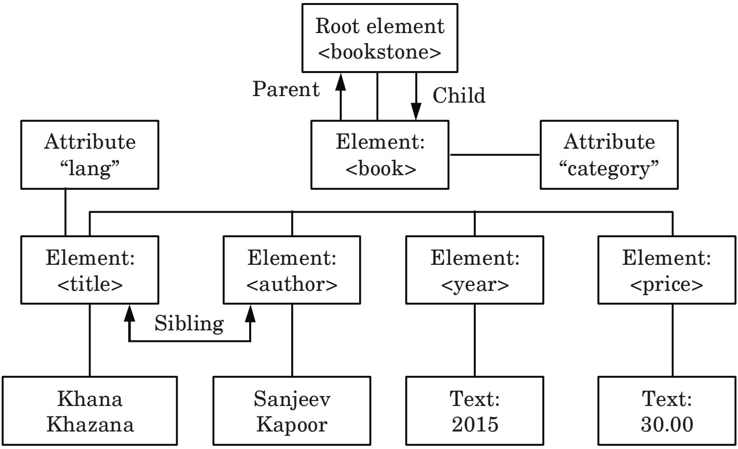


Fig. 2.

XML parser is useful in :

1. Loading the elements of XML document.
2. Accessing the elements of XML document.
3. Deleting the elements of XML document.
4. Changing the elements of XML document.

3. What is DTD ? What are the differences between external and internal DTD ? Use suitable example.

Ans. **DTD :**

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.

Difference :

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 : <!DOCTYPE root-element SYSTEM “file-name”> where file-name is the file with .dtd extension.	The syntax for internal DTD is : <!DOCTYPE root-element [element-declarations]> 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)>
<!ELEMENT phone (#PCDATA)> ]>
<address>
<name>Pratibha </name>
<company>Quantum</company>
<phone>(011)  123-4567</phone>
</address>
```

Example of external DTD :

```
<?xml version=“1.0” encoding="UTF-8" standalone=“no” ?>
<!DOCTYPE address SYSTEM “address.dtd”>
<address>
<name>Prabha 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)>
```

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.

Ans. **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.

JavaScript function for e-mail address validation :

```
<script type = “text/javascript”>
function validateEmail()
{
      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 ==“ ”));
      {
      alert(“Please Enter your Email ID”);
      emailID.focus();
      return false;
      }
      if (str.indexOf(“.”) == –1 | | str.indexOf(“.”) == 0 | |
      str.indexOf(“@”) ==–1 | | str.indexOf(“.”) == index_at
      | | str.indexOf(“@”,(index_at+1))!= –1) | |
      | | str.indexOf(“ ”))!= –1 | |
      {
            alert(“Invalid Email ID”);
            return false;
            }
            return true;
      }
</script>
```

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

Ans.

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.
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.

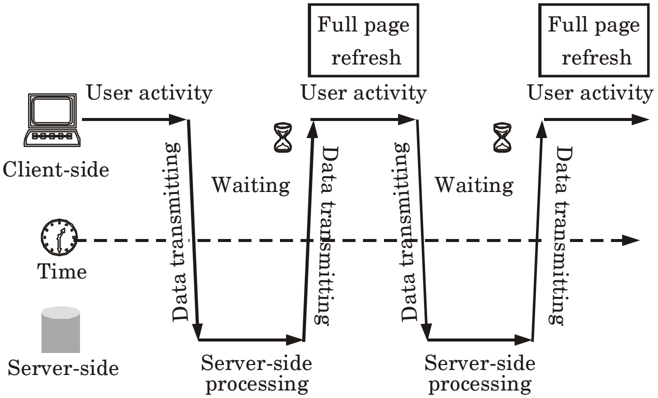


Fig. 3. Traditional web application communication flow.

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. 4.

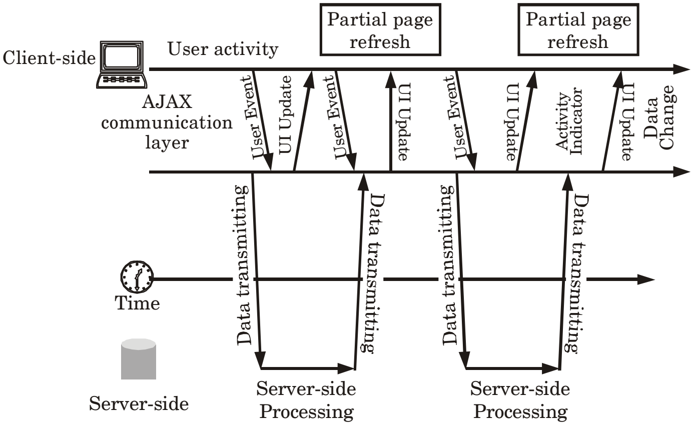


Fig. 4. AJAX style communication flow.

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.
4. Major application of AJAX is in login forms where user can enter their login details directly on the original page.

For example :

```
<!DOCTYPE html>
<html>
<style>
table,th,td {
border : 1px solid black;
border-collapse: collapse;
}
th,td {
padding: 5px;
}
</style>
<body>
<h1>The XMLHttpRequest Object</h1>
<form action="">
<select name="customers" onchange="showCustomer(this.value)">
<option value="">Select a customer:</option>
<option value="name1">Aakash Pandey</option>
<option value="name2">Mohan</option>
<option value="name3">Harshit Kumar</option>
</select>
</form>
<br>
```

```
<div id=“txtHint”>Customer info will be listed here...</div>
<script>
function showCustomer(str) {
 var xhttp;
if (str == “”) {
document.getElementById(“txtHint”).innerHTML = “”;
return;
}
xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById(“txtHint”).innerHTML=
this.responseText;
      }   };
xhttp.open(“GET”, “getcustomer.asp?q=”+str, true);
xhttp.send();
}
</script>
</body>
</html>
```

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

Ans. **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.

Life cycle of Servlet :

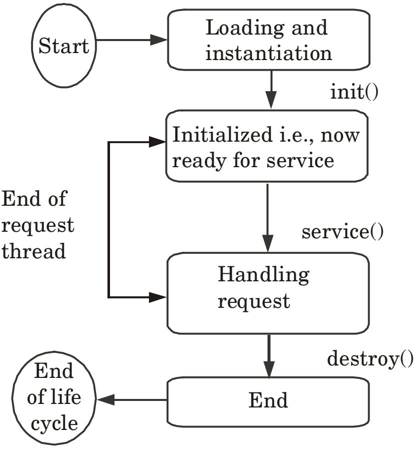


Fig. 5.

Stages of the Servlet life cycle :

1. **Loading a Servlet :**
   a. The first stage of the Servlet lifecycle involves loading and initializing the Servlet by the Servlet container.
   b. The Servlet container performs two operations in this stage :
      i. Loading : Loads the Servlet class.
      ii. Instantiation : Creates an instance of the Servlet. To create a new instance of the Servlet, the container uses the no-argument constructor.
2. **Initializing a Servlet :**
   a. After the Servlet is instantiated successfully, the Servlet container initializes the instantiated Servlet object.
   b. The container initializes the Servlet object by invoking the Servlet.init(ServletConfig) method which accepts ServletConfig object reference as parameter.
3. **Handling request :**
   a. After initialization, the Servlet instance is ready to serve the client requests.
   b. The Servlet container performs the following operations when the Servlet instance is located to service a request :
      i. It creates the ServletRequest and ServletResponse objects.
      ii. After creating the request and response objects it invokes the Servlet.service(ServletRequest, ServletResponse) method by passing the request and response objects.
4. **Destroying a Servlet :**
   a. When a Servlet container decides to destroy the Servlet, it performs the following operations,
      i. It allows all the threads currently running in the service method of the Servlet instance to complete their jobs and get released.
      ii. After currently running threads have completed their jobs, the Servlet container calls the destroy() method on the Servlet instance.
   b. After the destroy() method is executed, the Servlet container releases all the references of this Servlet instance so that it becomes eligible for garbage collection.

Characteristics of servlet :

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

Java servlets have following advantage over CGI and other API's :

i. **Platform independence :** Java servlets are pure Java program, so it is platform independent. It can run on any servlet enabled web server.

ii. **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.

iii. **Extensibility :** Java servlets are developed in Java which is robust, well-designed and object-oriented language which can be extended into new objects.

iv. **Safety :** Java servlet provides a very good safety features like memory management, exception handling features and emerged as a very powerful web server extension.

v. **Secure :** Servlets are server-side components, so it inherits the security provided by the web server.

7. What do you mean by JSP processing ? How JSP pages are handled ? Explain various JSP directives with suitable examples.

Ans. **JSP processing :**

1. 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.
2. JSP pages can be processed using JSP container only.
3. Following are the steps that need to be followed while processing the request for JSP page :

a. Client makes a request for required JSP page to the server.

b. The server must have JSP container so that JSP request can be processed.

c. On receiving request the JSP container searches and then reads the desired JSP page.

d. Then JSP page is converted to corresponding servlet. Basically any JSP page is a combination of template text and JSP element.

e. Every template text is translated into corresponding println statement.

JSP page handling :

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" %>
```

JSP directives :

Various JSP directives are :

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:
<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/>
...
```

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

Ans. **EJB architecture :**

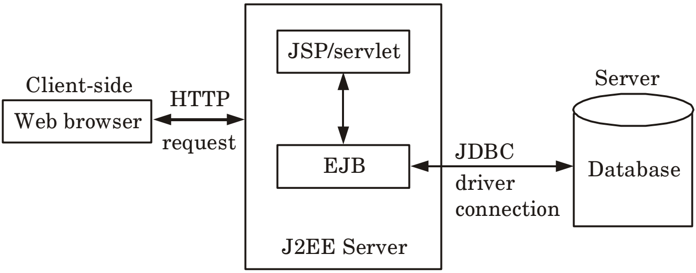


Fig. 6. Architecture.

The EJB architecture is an extension of web architecture.

Working of EJB architecture :

1. The client is working on a web browser.
2. There is a database server that hosts a database, like MySQL / Oracle.
3. The J2EE server machine is running on an application server.
4. The client interface is provided with JSP / Servlet.

5. The application server manages the relationships between the client and database.

Types of EJB :

1. **Entity bean :** Entity beans represent persistent data storage. Entity beans are used for modeling the business concept.
2. **Session bean :** Session beans are used for managing processes or tasks. Hence, session beans are used for managing activities.
3. **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 SimpleBean.jar”.

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.
2. Select SimpleBean.jar.
3. 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 java.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.

9. Describe the PHP life cycle and how does PHP score over ASP and JSP ?

Ans. **PHP life cycle :**

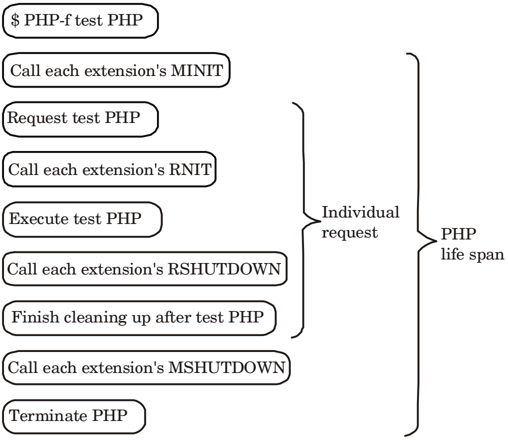

$ PHP-f test PHP
Call each extension's MINIT
Request test PHP
Call each extension's RNIT
Execute test PHP
Call each extension's RSHUTDOWN
Finish cleaning up after test PHP
Call each extension's MSHUTDOWN
Terminate PHP
Individual request
PHP life span

Fig. 7.

Explanation :

1. When we start Apache server, it starts the PHP interpreter.
2. PHP calls MINIT (Module Initialization) method of each extension. Each MINIT method initializes and defines a set of functions and classes.
3. Now, PHP layer set up an environment to execute the requested PHP page. It creates a symbol table which stores various variables used for executing this page.
4. PHP then calls RNIT (Request Initialization) method for each module. It is pre-appended to every PHP script before execution.
5. PHP execution is completed by reaching the end of the script.
6. Now, PHP starts the cleanup process. RSHUTDOWN method of every extension is called to destroy the symbol table.
7. Finally, when all requests have been made PHP calls the MSHUTDOWN method of every extension and free all persistent memory allocated during the MINIT cycle and PHP is terminated.

PHP over ASP and JSP :

1. PHP uses a blend of interpretation and compilation in order to provide the best mix of performance and flexibility to programmers.
2. PHP compiles our script down to a series of instructions (called opcodes) whenever it is accessed. These instructions are then executed one by one until the script terminates.
3. Text editor for PHP is available free of cost but test editor for ASP.Net *i.e.*, visual studies is not freely available.
4. PHP can run on any operating system but ASP code runs only on windows.
5. PHP hosting is cheaper than JSP hosting.
6. Furthermore, it provides very quick feedback during development. If we have an error somewhere in our file, PHP will refuse to compile the page until we have fixed the problem.

SECTION – C

Note : Attempt any **two** questions from this section.

(15 × 2 = 30)

10. Explain JDBC application architecture. What are the various types of JDBC drivers ? Write steps to connect database with the web application using JDBC.

Ans. **JDBC architecture :**

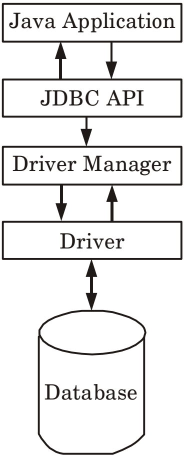


Fig. 8.

1. **Java application :** It is a standalone Java program which uses the JDBC API to get connected and perform operations on the database data.

2. **JDBC API :** It is a set of classes and interfaces used in a Java program for database operations. Java.sql and Javax.sql packages provide the necessary library support.
3. **Driver manager :** Java program uses DriverManager class to get the connection with the database.
4. **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.
5. **Database :** It is a collection of all enterprise data.

Types of JDBC driver :

1. **JDBC-ODBC bridge driver (Type 1 driver) :**
   a. These drivers are the bridge drivers such as JDBC-ODBC bridge.
   b. These drivers rely on an intermediary such as ODBC to transfer the SQL calls to the database.
   c. Bridge drivers often rely on native code, although the JDBC-ODBC library native code is part of the Java-2 virtual machine.
2. **Native API partly Java driver (Type 2 driver) :**
   a. A native API is partly a Java driver. It uses native C language library calls to translate JDBC to native client library.
   b. These drivers are available for Oracle, Sybase, DB2 and other client library based RDBMS.
   c. Type 2 drivers use native code and require additional permission to work in an Applet.
3. **JDBC net pure Java driver (Type 3 driver) :**
   a. JDBC net pure Java driver consists of JDBC and DBMS independent protocol driver.
   b. Here the calls are translated and sent to middle tier server through the socket.
   c. The middle tier contacts the database.
   d. Type 3 drivers call the database API on the server.
4. **Native protocol pure Java driver (Type 4 driver) :**
   a. A native protocol Java driver contains JDBC calls that are converted directly to the network protocol used by the DBMS server.
   b. This driver interacts directly with database.
   c. It does not require any native database library. So, it is also called thin driver.

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");

11. What is PHP ? Describe PHP session in detail. Explain the steps involved in connecting database in PHP.

Ans. PHP :

1. PHP is a Hypertext Preprocessor, server-side scripting language that is used to create dynamic web pages.
2. PHP is used to manage dynamic content, database, session tracking, even build e-commerce sites.
3. Syntax of PHP is shown as :

```
<?php
PHP Code!
?>
```

PHP session :

1. PHP session is an alternative way to make data accessible across the various pages of an entire website.
2. A session creates a file in a temporary directory on the server where registered session variables and their values are stored.
3. This data will be available to all pages on the site during that visit.
4. The location of the temporary file is determined by a setting in the php.ini file called session.save_path. Before using any session variable make sure that we have setup this path.
5. When a session is started following things happen :
   a. PHP first creates a unique identifier for that particular session which is a random string of 32 hexadecimal numbers such as 3c7foj34c3jj973hjkop2fc937e3443.
   b. A cookie called PHPSESSID is automatically sent to the user's computer to store unique session identification string.
   c. A file is automatically created on the server in the designated temporary directory and bears the name of the unique identifier prefixed by sess_ *i.e.*, sess_3c7foj34c3jj973hjkop2fc937e3443.

Steps involved in connecting database in PHP :

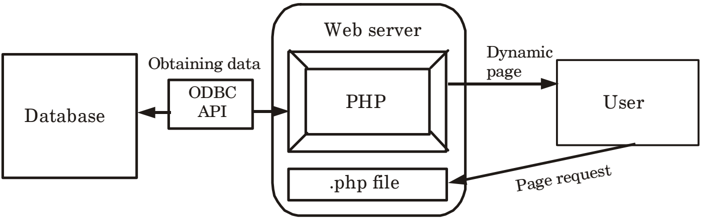


Fig. 9. The PHP database with ODBC API architecture.

Fig. 9 depicts the following :

The visitor's :

Web browser requests a webpage via (<filename>.php) using a URL.

The Web Server (Apache or IIS) :

1. Recognizes that the requested file is a PHP script.
2. The web server therefore interprets the file using its PHP engine.

PHP :

1. Processes the page request.
2. Fetches data from database/tables using the ODBC API.

ODBC API : Interacts with the specific DB engine requested by the page.

Database : Responds by sending the requested content to the PHP script.

PHP : The PHP script stores the content into one or more PHP variables.

12. Write down the short note on the following :

a. COM/DCOM

Ans. 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 the 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 is a model as COM but it is specially designed for distributed application.
4. DCOM, which originally was called "Network OLE" extends Microsoft's COM, and provides the communication substrate under Microsoft's COM+ application server infrastructure.
5. DCOM was a major competitor to CORBA.

b. Tomcat server

Ans.

1. Tomcat is a web server and servlet container that is used to deploy and serve Java web application.
2. Tomcat server is a Java-capable HTTP server, which could execute special Java programs known as "Java servlet" and "Java Server Pages".
3. Tomcat can operate as a standalone web server.
4. It can operate as an out-of-process servlet container for some web servers, such as Apache.
5. For other web servers, such as IIS (Internet Information Services), it can operate as an in-process servlet container.
6. Tomcat server runs on a specific TCP post from a specific IP address.
7. The default TCP port number for HTTP protocol is 80, which is used for the production HTTP server.
8. To test HTTP server, we can choose any unused port number between 1024 and 65535.
9. There are two important environment variables to set before running Tomcat :

a. Set CATALINA_HOME to the root of Tomcat directory.

b. Set JAVA_HOME to the root directory of Java JDK or JRE.

c. CORBA

Ans.

1. CORBA stands for Common Object Request Broker Architecture. It is a standard defined by Object Management Group (OMG). It allows a communication between the systems that are present on the different platforms.
2. It is an architecture and specification for creating, distributing and managing distributed program objects in a network.
3. Part of the CORBA standard is the Interface Definition Language (IDL), which is an implementation-independent language for describing the interfaces of remote objects.

☺☺☺

B. Tech.

(SEM. V) ODD SEMESTER THEORY EXAMINATION, 2016-17

WEB TECHNOLOGY

Time : 3 Hours **Max. Marks : 100**

Section-A

1. Answer all parts. All parts carry equal marks. Write answer of each part in short. **(2 × 10 = 20)**

a. **How do you call a package ?**

b. **Mention the usage of private access specifier.**

c. **What is HTML ?**

d. **What is selector ? Mention its types.**

e. **How a JavaScript works ?**

f. **Differentiate between dynamic web pages and static web pages.**

g. **What is Java Server Pages technology ?**

h. **Compare servlet with CGI.**

i. **What is the usage of CSS ?**

j. **Define ODBC.**

Section-B

2. Attempt any **five** questions from this section. **(10 × 5 = 50)**

a. **Briefly explain the internet concepts.**

b. **Compare and contrast Java and JavaScript.**

c. **Design a HTML form for a railway reservation system.**

d. **Explain in detail the Tomcat server.**

e. **What do you understand by ASP ? What are its advantages and disadvantages ?**

f. **Explain COM and DCOM in detail.**

g. **Describe the PHP life cycle and how does PHP score over ASP and JSP ?**

Section-C

Note : Attempt any two questions from this section. (15 × 2 = 30)

3. **Elucidate the features of java.**

4. **What is XML ? Discuss the significance of XML. How is XML different from HTML ? Explain the process of parsing. How are they useful ?**

5. **Write short notes on :**

a. **AJAX** b. **EJB architecture** c. **AWT**

☺☺☺

SOLUTION OF PAPER (2016-17)

Section-A

1. Answer all parts. All parts carry equal marks. Write answer of each part in short. **(2 × 10 = 20)**

a. How do you call a package ?

Ans. There are two ways to call the package :

1. Import packagename; 2. Import package.classname;

b. Mention the usage of private access specifier.

Ans. Usage of private access specifier is :

1. To access the private methods and fields only within the same class to which the methods and fields belong.
2. To form the symbolic relationship ("is a") between base and derived class.

c. What is HTML ?

Ans. HTML (Hypertext Markup Language) is a set of special codes that can be embedded in text to add formatting and linking information.

d. What is selector ? Mention its types.

Ans. A selector is a string that identifies what elements the corresponding rule is applied.

Types of selectors :

1. Simple selectors
2. Combinator selectors
3. Pseudo-class selectors
4. Pseudo-elements selectors
5. Attribute selectors

e. How a JavaScript works ?

Ans.

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

f. Differentiate between dynamic web pages and static web pages.

Ans.

S. No.	Dynamic web page	Static web page
1.	Dynamic web pages are behavioral and have the capacity to produce distinctive content for different visitors.	Static web pages will remain same for the time until and unless someone changes it manually.
2.	Dynamic pages are complex to construct.	Static web pages are simple to construct.

g. What is Java Server Pages technology ?

Ans. Java Server Pages (JSPs) are simple and powerful technology used to generate dynamic HTML documents on the server-side.

h. Compare servlet with CGI.

Ans.

S. No.	Servlet	CGI
1.	It is written in Java.	It is written in C, C++, Visual Basic.
2.	It is thread based.	It is process based.
3.	It is platform dependent.	It is platform independent.
4.	It delivers non-scalable application.	It delivers scalable application.

i. What is the usage of CSS ?

Ans. CSS is used to enhance the features of web pages like formatting styles and thereby increase the visual appearance of web pages. CSS helps in faster downloading of the pages.

j. Define ODBC.

Ans. Open Database Connectivity (ODBC) is an open standard API for accessing a database. By using ODBC statements in a program, we can access files in a number of different databases, including MS Access, DB2, Excel, and Text.

Section-B

2. Attempt any **five** questions from this section. **(10 × 5 = 50)**

a. Briefly explain the internet concepts.

Ans.

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, 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.

b. Compare and contrast Java and JavaScript.

Ans. **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.

c. Design a HTML form for a railway reservation system.

Ans.

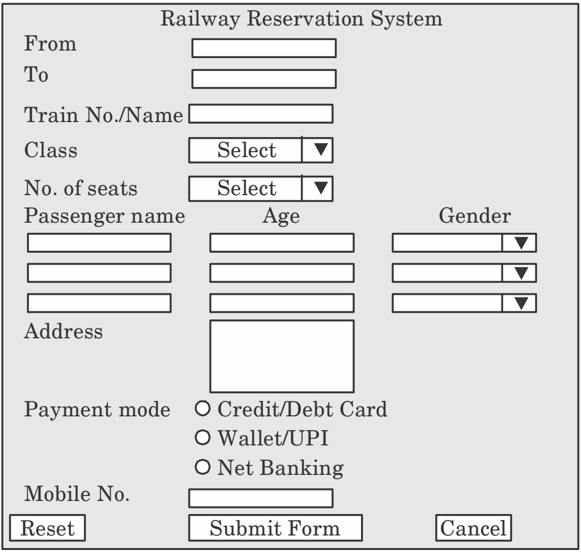
```
<html>
<head>
<script type = “text/JavaScript” src = “validate.js”></script>
</head>
<body>
<form action = “#” name = “RailwayReservationSystem”
onsubmit = “return(validate( ) ) ;”>
<table cellpadding = “2” width = “20%” bgcolor = “99FFFF”
align = “center” cellspacing = “2”>
```

```
<tr>
<td colspan = “2”>
<center><font size = “4”><b>Railway Reservation System</b></
font>< /center>
</td>
</tr>
<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>
</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>
<tr><td>Age</td></tr>
<tr><td>Gender</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 = “P2”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”>Wallet
UPI</td>
<td><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>
<td>
<input type=“Cancel” value=“Cancel”>
</td>
</tr>
</table>
</form>
```

```
</body>
</html>
```

Output :



Railway Reservation System

From ▭

To ▭

Train No./Name ▭

Class Select ▼

No. of seats Select ▼

Passenger name Age Gender

▭ ▭ ▼

▭ ▭ ▼

▭ ▭ ▼

Address ▭

Payment mode ○ Credit/Debt Card

○ Wallet/UPI

○ Net Banking

Mobile No. ▭

Reset Submit Form Cancel

d. **Explain in detail the Tomcat server.**

Ans.

1. Tomcat is a web server and servlet container that is used to deploy and serve Java web application.
2. Tomcat server is a Java-capable HTTP server, which could execute special Java programs known as "Java servlet" and "Java Server Pages".
3. Tomcat can operate as a standalone web server.
4. It can operate as an out-of-process servlet container for some web servers, such as Apache.
5. For other web servers, such as IIS (Internet Information Services), it can operate as an in-process servlet container.
6. Tomcat server runs on a specific TCP post from a specific IP address.
7. The default TCP port number for HTTP protocol is 80, which is used for the production HTTP server.
8. To test HTTP server, we can choose any unused port number between 1024 and 65535.
9. There are two important environment variables to set before running Tomcat :

a. Set CATALINA_HOME to the root of Tomcat directory.

b. Set JAVA_HOME to the root directory of Java JDK or JRE.

e. **What do you understand by ASP ? What are its advantages and disadvantages ?**

Ans.

1. ASP is a technology used for developing dynamic websites by letting a page author include logic, such as VBScript and JavaScript code, in regular web pages to generate the dynamic parts.
2. ASP uses server-side scripting to dynamically produce web pages that are not affected by the type of browser, the website visitor is using.
3. When an ASP page is requested, the code in the page is executed by the server.
4. The result is inserted into the page, and the combination of the static and dynamic content is sent to the browser.
5. ASP pages have the extension (.asp).

Advantages of ASP :

1. It provides three-tier architecture.
2. Provide easy way to deploy the new applications and maintenance.
3. Consistent programming model.
4. Direct support for security.
5. Easy to learn and install.

Disadvantages of ASP :

1. Interpreted and loosely-typed code.
2. Mix layout (HTML) and logic (scripting code).
3. Limited development and debugging tools.
4. No real state management.
5. Update files only when server is down.
6. Obscure configuration settings.

f. Explain COM and DCOM in detail.

Ans. **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 the 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 is a model as COM but it is specially designed for distributed application.
4. DCOM, which originally was called “Network OLE” extends Microsoft’s COM, and provides the communication substrate under Microsoft’s COM+ application server infrastructure.
5. DCOM was a major competitor to CORBA.

g. Describe the PHP life cycle and how does PHP score over ASP and JSP ?

Ans. PHP life cycle :

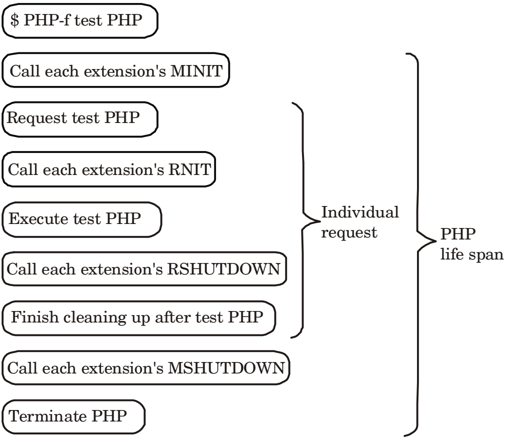

$ PHP-f test PHP
Call each extension's MINIT
Request test PHP
Call each extension's RNIT
Execute test PHP
Call each extension's RSHUTDOWN
Finish cleaning up after test PHP
Call each extension's MSHUTDOWN
Terminate PHP
Individual request
PHP life span

Fig. 1.

Explanation :

1. When we start Apache server, it starts the PHP interpreter.
2. PHP calls MINIT (Module Initialization) method of each extension. Each MINIT method initializes and defines a set of functions and classes.
3. Now, PHP layer set up an environment to execute the requested PHP page. It creates a symbol table which stores various variables used for executing this page.
4. PHP then calls RNIT (Request Initialization) method for each module. It is pre-appended to every PHP script before execution.
5. PHP execution is completed by reaching the end of the script.
6. Now, PHP starts the cleanup process. RSHUTDOWN method of every extension is called to destroy the symbol table.
7. Finally, when all requests have been made PHP calls the MSHUTDOWN method of every extension and free all persistent memory allocated during the MINIT cycle and PHP is terminated.

PHP over ASP and JSP :

1. PHP uses a blend of interpretation and compilation in order to provide the best mix of performance and flexibility to programmers.

2. PHP compiles our script down to a series of instructions (called opcodes) whenever it is accessed. These instructions are then executed one by one until the script terminates.
3. Text editor for PHP is available free of cost but test editor for ASP.Net *i.e.*, visual studies is not freely available.
4. PHP can run on any operating system but ASP code runs only on windows.
5. PHP hosting is cheaper than JSP hosting.
6. Furthermore, it provides very quick feedback during development. If we have an error somewhere in our file, PHP will refuse to compile the page until we have fixed the problem.

Section-C

Note : Attempt any two questions from this section. (15 × 2 = 30)

3. Elucidate the features of java.

Ans. The basic features of Java are as follows :

1. **Simple :** Java is easy to learn.
2. **Object-oriented :** All the functions are defined inside the classes.
3. **Platform independent :** It is platform independent programming language because it compiles to byte-code. This byte code is interpreted by the Virtual Machine (JVM) on any platform.
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. So it is more secure.
6. **Portable :** Java is portable because it facilitates us to carry the Java bytecode to any platform. It does not require any implementation.
7. **Multithreaded :** Java programs are capable for easy to set up multiple threads of execution and coordinate parallel processes.
8. **Dynamic :** Java is a dynamic language. It supports dynamic loading of classes. It means classes are loaded on demand. It also supports functions from its native languages, i.e., C and C++.
9. **Garbage collection :** Java has an automatic garbage collector which releases the objects which are not in use from long time.

4. What is XML ? Discuss the significance of XML. How is XML different from HTML ? Explain the process of parsing. How are they useful ?

Ans.

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>
<price>29.99</price>
</book>
<book category=“web” cover=“paperback”>
<title lang=“en”>Learning XML</title>
<author>Erik T. Ray</author>
<year>2013</year>
<price>39.95</price>
</book>
</bookstore>
```

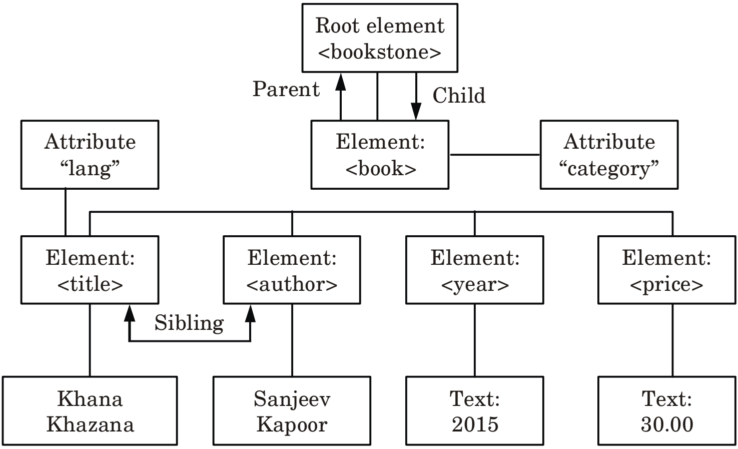


Fig. 2.

XML parser is useful in :

1. Loading the elements of XML document.
2. Accessing the elements of XML document.
3. Deleting the elements of XML document.
4. Changing the elements of XML document.

5. Write short notes on :

a. AJAX **b. EJB architecture** **c. AWT**

Ans.

a. AJAX :

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. 2.
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.

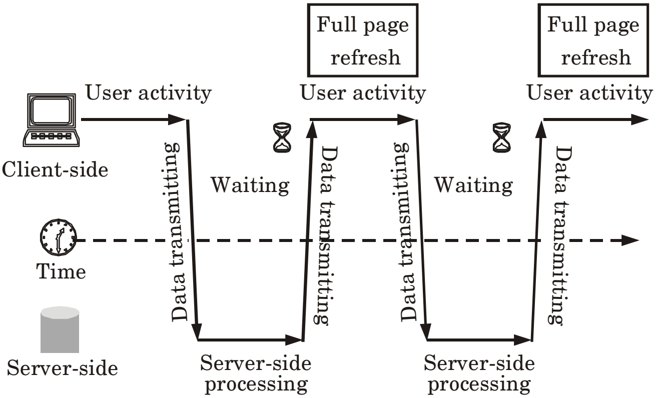


Fig. 3. Traditional web application communication flow.

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. 4.

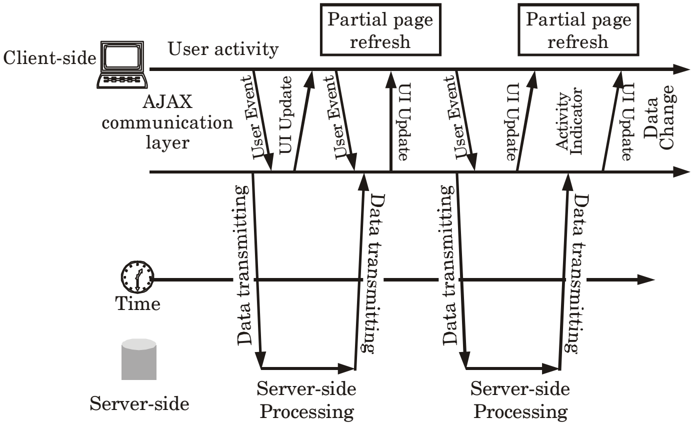


Fig. 4. AJAX style communication flow.

b. EJB architecture :

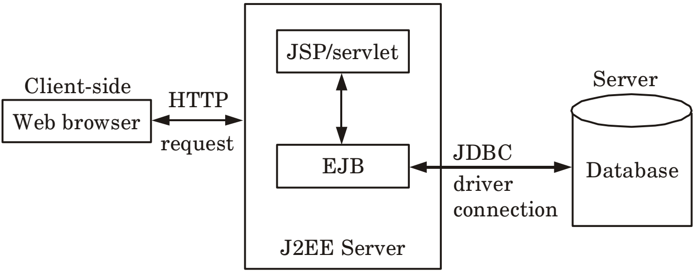


Fig. 5. Architecture.

The EJB architecture is an extension of web architecture.

Working of EJB architecture :

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

c. AWT :

1. The AWT stands for Abstract Window Toolkit.
2. AWT is a library of class which provides GUI tools to develop GUI application and applet.
3. It provides many classes for programmers to use. It is the connection between our application and the native GUI.
4. 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 :
   a. **Containers class :** Frame, Dialog, Panel, Applet etc.
   b. **Components class :** TextField, Button, Checkbox, Scrollbar, Label, List etc.
   c. **Custom graphics class :** Colour, Font, Dimensions etc.

☺☺☺

B. Tech.

(SEM. V) ODD SEMESTER THEORY EXAMINATION, 2017-18

WEB TECHNOLOGY

Time : 3 Hours **Max. Marks : 100**

Note : 1. Attempt **all** questions. If require any missing data; then choose suitably.

2. Illustrate with program and comments if required.

Section-A

1. Attempt **all** questions in brief : **(2 × 10 = 20)**

a. Explain the anchor and table tag in HTML.

b. What are forms and how they are created in HTML ?

c. Explain the difference between id and class selector in CSS.

d. Explain well formed and valid XML.

e. What is a web project ?

f. Explain client-server architecture with diagram.

g. Write various types of drivers available in java for database handling.

h. Why filters are used in PHP ?

i. What is EJB ? Explain its types.

j. What are JAR files ? Explain its advantage. Write the command for creating JAR file.

Section B

2. Attempt any **three** of the following : **(10 × 3 = 30)**

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

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

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

d. **What is AJAX ? Explain its advantage and its working. Explain with example.**

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

Section-C

3. Attempt any **one** part of the following : **(10 × 1 = 10)**

a. **What are JSP directives ? Explain various types of directives with example.**

b. **Explain implicit objects available in JSP with example.**

4. Attempt any **one** part of the following : **(10 × 1 = 10)**

a. **What are standard actions in JSP ? Illustrate with example.**

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

5. Attempt any **two** parts of the following : **(5 × 2 = 10)**

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

b. **Define HTML DOM.**

c. **Explain filters in PHP. Why they are used ? Explain with proper example.**

6. Attempt any **two** parts of the following : **(5 × 2 = 10)**

a. **Explain various protocols governing web. Also explain web team.**

b. **What is PHP ? Describe PHP session in detail. Explain the steps involved in connect PHP page to MySQL database.**

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

7. Attempt any **one** part of the following : **(10 × 1 = 10)**

a. **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.

b. **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 ?**

☺☺☺

SOLUTION OF PAPER (2017-18)

Note : 1. Attempt **all** questions. If require any missing data; then choose suitably.
2. Illustrate with program and comments if required.

Section-A

1. Attempt **all** questions in brief : **(2 × 10 = 20)**

a. Explain the anchor and table tag in HTML.

Ans. Anchor tag :

1. The HTML anchor tag defines a hyperlink that links one page to another page.
2. The “href” attribute is used to specify a target for the anchor tag.
3. The syntax of HTML anchor tag :

<a href=‘......’> Link Text</a>

Table tag :

A table is a two dimensional matrix, consisting of rows and columns.

Example :

```
<table><tr>
<td> row 1</td>
<td> row 1</td>
</tr>
</table>
```

b. What are forms and how they are created in HTML ?

Ans. An HTML form is a section of document containing normal content, markup, special elements called controls (checkboxes, radio buttons, menus etc.) and labels on those controls.

The HTML <form> tag is used to create an HTML form and it has following syntax :

<form action = “script URL” method = “GET | POST”>
Form elements like input, text area etc.
</form>

c. Explain the difference between id and class selector in CSS.

Ans.

S. No.	Id selector	Class selector
1.	It is used to specify a style for a unique element.	It is used to specify a style for a group of element.
2.	The syntax for id selector is : #idselector_name {property:value;}	The syntax of class selector is : class_selector_name {property:value;}

d. Explain well formed and valid XML.

Ans. An XML document with correct syntax is called well formed XML. An XML document is valid if it is validated against a DTD.

e. What is a web project ?

Ans. A web project is the process of developing and creating different types of websites that are implemented on the internet.

f. Explain client-server architecture with diagram.

Ans. In client-server architecture, the clients are programs running on remote machines that communicate with a program called the server that runs on a single site and responds to requests from many clients.

g. Write various types of drivers available in java for database handling.

Ans.

1. JDBC-ODBC bridge driver
2. Native API driver
3. Network protocol driver
4. JDBC Net driver

h. Why filters are used in PHP ?

Ans. The PHP filter is used to validate and filter the data. When the data source contains unknown or foreign data then the PHP filter is used.

PHP filter can be classified as :

a. Validating filters
b. Sanitizing filters

i. What is EJB ? Explain its types.

Ans. Enterprise Java Bean (EJB) is the server-side and platform independent Java application programming interface (API) for Java Platform Enterprise Edition (Java EE). EJB is used to simplify the development of large distributed applications.

Types of EJB are :

1. Entity Bean
2. Session Bean
3. Message driven Bean

j. What are JAR files ? Explain its advantage. Write the command for creating JAR file.

Ans. A JAR (Java ARchive) file is a file that contains compressed version of class files, audio files, image files or directories.

Advantages of JAR file :

a. Security
b. Decreased download time

c. Compression

To create a JAR file, we can use following command :

```
jar cf jarfilename inputfiles
```

Section B

2. Attempt any **three** of the following : **(10 × 3 = 30)**

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

Ans. CSS :

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>.

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>
```

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

Ans. 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.

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. 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.

b. The catch block can have one or more statements that are necessary to process the exception.

c. The catch statement is passed as a single parameter, which is reference to the type of exception object thrown by the try block.

d. 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 :

a. 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.

b. Finally block can be used to handle any exception generated within a try block.

c. It may be added immediately after the try block or after the last catch block as follows :

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

4. **Throw :**
   a. Java supports "throw keyword" which is used if we want to throw our own exceptions.
   b. We can do this by using the keyword throw as follows :

```
throw new Throwable_subclass;
```

No, it is not essential to catch all type of exceptions.

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.ArithmeticException: / by zero
finally block is always executed
rest of the code...
```

c. What are packages in java ? How a user defined package is created in java, explain with example ?

Ans.

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);
}
}
```

d. What is AJAX ? Explain its advantage and its working. Explain with example.

Ans. **AJAX :**

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. 1.

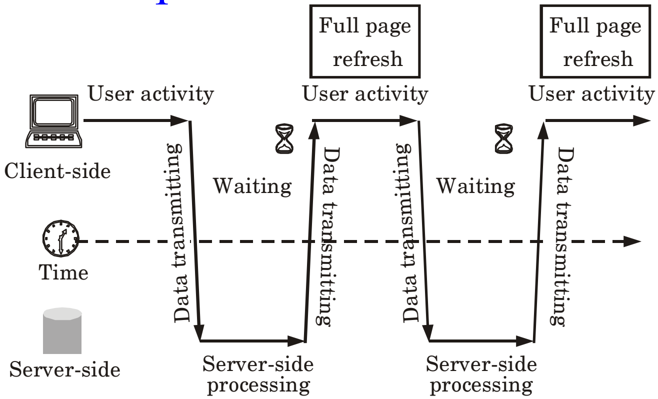


Fig. 1 Traditional web application communication flow.

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.
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. 2.

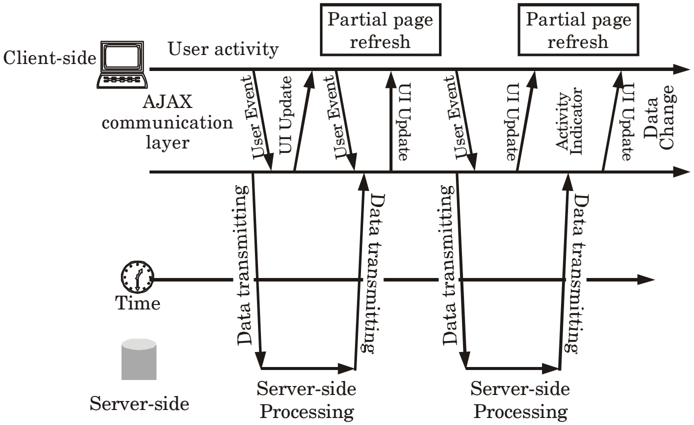


Fig. 2. AJAX style communication flow.

Advantages of AJAX :

1. Reduces the server traffic and increases the speed.
2. Ajax is responsive and time for data transfer is also less.
3. Form validation
4. Bandwidth usage can be reduced.
5. Asynchronous calls can be made which reduces the time for data arrival.

Working of AJAX : XMLHttpRequest object plays an important role as AJAX communicates with the server using XMLHttpRequest object.

1. User sends a request from the UI and a JavaScript call goes to XMLHttpRequest object.
2. HTTP request is sent to the server by XMLHttpRequest object.
3. Server interacts with the database using JSP, PHP, Servlet, ASP.net etc.
4. Data is retrieved.
5. Server sends XML data or JSON data to the XMLHttpRequest callback function.
6. HTML and CSS data is displayed on the browser.

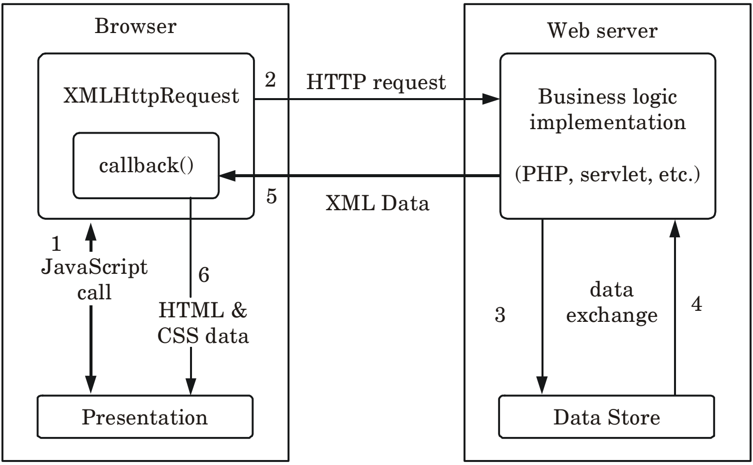


Fig. 3.

For 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>
```

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

Ans.

DTD :

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.

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 markup declarations that provide a grammar for a class of documents.
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.	**Example :** <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”/>	**Example :** <!DOCTYPE not [<!ELEMENT not (to, from, heading, body)> <!Element to (# PCDATA) > <!Element from (# PCDATA)> <! Element heading (?)> <!Element body (#PCDATA)]>

Section-C

3. Attempt any **one** part of the following : **(10 × 1 = 10)**

a. What are JSP directives ? Explain various types of directives with example.

Ans. 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:
<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/>
...
```

b. **Explain implicit objects available in JSP with example.**

Ans. Different types of implicit objects in JSP are :

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">
<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>:<%=config.getInitParameter("User") %><br><br>
<%-- application object example --%>
<strong>User context param value</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() %>
</body>
</html>
```

4. Attempt any **one** part of the following : **(10 × 1 = 10)**

a. What are standard actions in JSP ? Illustrate with example.

Ans. 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.

6. <jsp:body> :

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/gets 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>
<head><title>jsp:text action</title></head>
<body>
<books><book><jsp:text>
Welcome to JSP Programming
</jsp:text></book></books>
</body>
</html>
```

b. Explain the life cycle of servlet. Also write a servlet for displaying a string “HELLO WORLD!”

Ans. Life cycle of servlet :

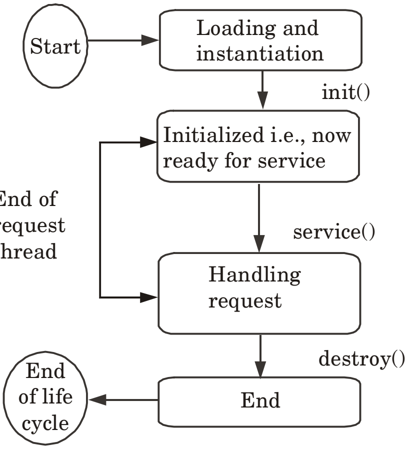


Fig. 4.

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.
}
}
```

5. Attempt any **two** parts of the following : **(5 × 2 = 10)**

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

Ans. 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 :

```
<script type="text/javascript">
function validateform()
{
var name = document.myform.name.value;
var password = document.myform.password.value;
var confirmpassword = document.myform.password2.value;
var email = document.myform.email.value;
# Username validation
if (name == null || name == ""){
```

```
alert(“Name can’t be blank”);
return false;
}else if(password.length<6){
alert(“Password must be at least 6 characters long.”);
return false;
          }
}
# Retype password validation
if (password == confirmpassword){
return true;
          }
else{
alert(“password must be same!”);
return false;
}
}
# Email validation
var emailErr = True;
if(email == “”) {
printError(“emailErr”, “Please enter your email address”);
          } else {
// Regular expression for basic email validation
          var regex = /^\S+@\S+\.\S+$/;
          if(regex.test(email) === false) {
            printError(“emailErr”, “Please enter a valid email address”);
          } else{
          printError(“emailErr”, “”);
          emailErr = false;
          }
          }
</script>
<body>
<form name=“myform” method=“post” onsubmit=“return
validateform()”>
Username:   <input type=“text” name=“name”><br/>
Password :   <input type=“password” name=“password” /><br/>
Re-enter Password:   <input type=“password” name=“password2”/
><br/>
Email :   <input type=“text” name=“email”>
<div class=“error” id=“emailErr”></div>
<input type=“submit” value=“Register”>
</form>
</body>
</html>
```

b. Define HTML DOM.

Ans.

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>
<name> "Web application development with XML and Java"
</name>
</book>
</goods>
</book-order>
```

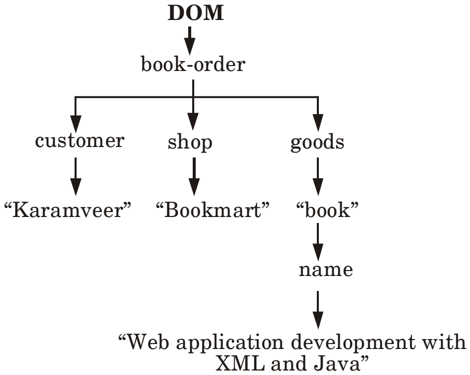


Fig. 5.

c. Explain filters in PHP. Why they are used ? Explain with proper example.

Ans.

1. A PHP filter is used to validate and filter data coming from insecure sources.
2. To test, validate and filter user input or custom data is an important part of any web application.

3. The PHP filter extension is designed to make data filtering easier and quicker.
4. There are two kinds of filter :

   a. Validating filter :

   i Used to validate user input.

   ii. Strict format rules (like URL or e-mail validating).

   iii. Returns the expected type on success or FALSE on failure.

   b. Sanitizing filter :

   i. Used to allow or disallow specified characters in a string.

   ii. No data format rules.

   iii. Always return the string.

Use of filter :

By using filters, we can be sure our application gets the correct input type.

For example :

In this we validate the output from a form.

We assume that the input data exists.

Then, we filter the input data using the filter_input() function.

In program, the input variable "email" is sent to the PHP page :

```
<?php
if(!filter_has_var(INPUT_GET, "email"))
{
echo("Input type does not exist")
}
else
{
if (!filter_input(INPUT_GET, "email"),
FILTER_VALIDATE_EMAIL))
{
echo ("E-Mail is not valid");
}
else
{
echo ("E-Mail is valid");
}
}
?>
```

6. Attempt any **two** parts of the following : **(5 × 2 = 10)**

a. Explain various protocols governing web. Also explain web team.

Ans. Protocols governing web :

1. HTTP (Hypertext Transfer Protocol) :

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 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 (Internet Control Message Protocol) :

- a. 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 (Routing Information Protocol) :

- a. 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 (Open Shortest Path First) :

- a. OSPF is classified as an Interior Gateway Protocol (IGP).
- b. 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.

5. TCP/IP :

- a. TCP/IP stands for Transmission Control Protocol / Internet Protocol.
- b. It 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. UDP :

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

Web team :

1. Web team is a group of various technical experts in a developing site from coding the page to maintain the web server.
2. The ideal web team consists of two sub-teams :
   - a. The client-side specialists, who create an attractive, clear front-end.
   - b. The server-side specialists, who create a smoothly operating back-end.

b. **What is PHP ? Describe PHP session in detail. Explain the steps involved in connect PHP page to MySQL database.**

Ans. **PHP :**

1. PHP is a Hypertext Preprocessor, server-side scripting language that is used to create dynamic web pages.
2. PHP is used to manage dynamic content, database, session tracking, even build e-commerce sites.
3. Syntax of PHP is shown as :

```
<?php
PHP Code!
?>
```

PHP session :

1. PHP session is an alternative way to make data accessible across the various pages of an entire website.
2. A session creates a file in a temporary directory on the server where registered session variables and their values are stored.
3. This data will be available to all pages on the site during that visit.
4. The location of the temporary file is determined by a setting in the php.ini file called session.save_path. Before using any session variable make sure that we have setup this path.
5. When a session is started following things happen :
   a. PHP first creates a unique identifier for that particular session which is a random string of 32 hexadecimal numbers such as 3c7foj34c3jj973hjkop2fc937e3443.
   b. A cookie called PHPSESSID is automatically sent to the user's computer to store unique session identification string.
   c. A file is automatically created on the server in the designated temporary directory and bears the name of the unique identifier prefixed by sess_ *i.e.*, sess_3c7foj34c3jj973hjkop2fc937e3443.

Steps involved in connecting database in PHP :

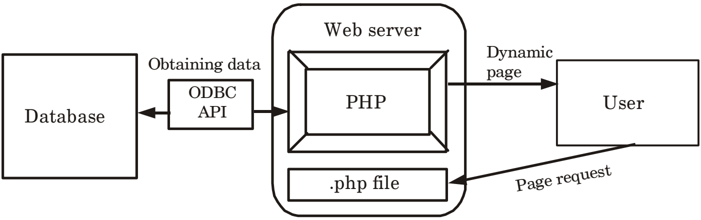


Fig. 6. The PHP database with ODBC API architecture.

Fig. 6 depicts the following :

The visitor's :

Web browser requests a webpage via (<filename>.php) using a URL.

The Web Server (Apache or IIS) :

1. Recognizes that the requested file is a PHP script.
2. The web server therefore interprets the file using its PHP engine.

PHP :

1. Processes the page request.
2. Fetches data from database/tables using the ODBC API.

ODBC API : Interacts with the specific DB engine requested by the page.

Database : Responds by sending the requested content to the PHP script.

PHP : The PHP script stores the content into one or more PHP variables.

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

Ans. **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 :

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 root-element [element-declarations]>

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”>

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)>
<!ELEMENT phone (#PCDATA)> ]>
<address>
<name>Pratibha </name>
<company>Quantum</company>
<phone>(011)  123-4567</phone>
</address>
```

Example of external DTD :

```
<?xml version=“1.0” encoding="UTF-8" standalone=“no” ?>
<!DOCTYPE address SYSTEM “address.dtd”>
<address>
<name>Prabha 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)>
```

7. Attempt any **one** part of the following : **(10 × 1 = 10)**

a. 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.

Ans. 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”);
PreparedStatement 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>”);
}
ut.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>
```

b. 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 ?

Ans. **Java Beans :**

1. Java Beans are classes which encapsulate several objects into a single object.
2. It helps in accessing the objects from multiple places.
3. It is a portable, platform independent model written in Java.

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 constructor with no argument.
4. It can be manipulated visually in a builder tools.

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:text> : This action is used to encapsulate template text that should be used in JSP pages written as XML documents.

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/gets 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:text> action :

```
<jsp:text><![CDATA[<!DOCTYPE html PUBLIC “-//W3C//DTD
XHTML 1.0 Strict//EN”
“DTD/xhtml1-strict.dtd”>]]></jsp:text>
<head><title>jsp:text action</title></head>
<body>
<books><book><jsp:text>
Welcome to JSP Programming
</jsp:text></book></books>
</body>
</html>
```

☺☺☺

B. Tech.

(SEM. V) ODD SEMESTER THEORY EXAMINATION, 2018-19

WEB TECHNOLOGIES

Time : 3 Hours **Max. Marks : 70**

Note : Notes : Assume any missing data.

Section-A

1. Attempt **all** questions in brief : **(2 × 7 = 14)**

a. What is bytecode ?

b. Define constructor.

c. What is an instance variable ?

d. When is it appropriate to use frames ?

e. What is the use of alternative text in image mapping ?

f. What are the two major protocols for accessing e-mail from servers ?

g. What do you mean by checked exceptions ?

Section-B

2. Attempt any **three** of the following : **(7 × 3 = 21)**

a. Create an html page named as "Table.html" to display your class time table.

i. Provide the title as Time Table.

ii. Provide various colour options to the cells (Highlight the lab hours and elective hours with different colours).

b. Compare Java and JavaScript. Write a JavaScript program to define a user defined function for sorting the values in an array.

c. What is the difference between applet and application ? How is Java strongly associated with internet ? Draw a flowchart to show various Java tools are used in application development.

d. Compare JSP and Servlet. Explain the life cycle of a JSP page with a suitable diagram. Also list any five action tags used in JSP.

e. What are the uses of layout managers ? Give the name of classes that represents the layout managers. Explain any five layout managers.

Section-C

3. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. Explain the HTTP protocol. Mention three basic features of HTTP that make HTTP a simple but powerful protocol. Give its architecture.

b. What is XML ? Create a XML document of 10 students of final CSE. Add their roll number, marks obtained in 5 subjects, total marks and percentage. Save this XML document at the server, write a program that accepts student's roll number as input and returns the students marks, total percentage by taking student information for XML document.

4. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. Discuss how frames play a big role in advertising on web. What roles do form play in making web page dynamic.

b. What are exceptions and how they are handled ? Explain with an example. How we define a try and catch block ? Is it essential to catch all types of exceptions ?

5. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. Create an html page named as "String_Math.html" and within the script tag define some string variables and use different string function to demonstrate the use of the predefined functions. Do the same for the math function.

b. What are the advantages and drawback of applet ? Write a Java program to create an applet for calculator and also perform event handling on each button.

6. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. What do you mean by CSS ? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system. Mention how can you integrate CSS on a web page?

b. What is difference between session and cookies ? Write a servlet program for servlet login and logout using cookies.

7. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. What is JDBC ? Explain the drivers used in JDBC. Write a JDBC program for insert and display the record of employees using prepared statement.

b. What are XML parsers ? Explain the types of parsers with their advantages and disadvantages.

☺☺☺

SOLUTION OF PAPER (2018-19)

Note : Notes : Assume any missing data.

Section-A

1. Attempt **all** questions in brief : **(2 × 7 = 14)**

a. What is bytecode ?

Ans. Bytecode is a program code that has been compiled from source code into low-level code and designed for a software interpreter.

b. Define constructor.

Ans. A constructor is a special type of subroutine called to create an object. It prepares the new object for use, often accepting arguments that the constructor uses to set required member variables.

c. What is an instance variable ?

Ans. An instance variable is a variable defined in a class, for which each instantiated object of the class has a separate copy, or instance. An instance variable is similar to a class variable.

d. When is it appropriate to use frames ?

Ans. When user wants to view multiple documents within a single web page, then it is appropriate to use frames.

e. What is the use of alternative text in image mapping ?

Ans. The alternative text in image mapping is use to provides alternative information for an image if image cannot be displayed.

f. What are the two major protocols for accessing e-mail from servers ?

Ans. Post Office Protocol version 3 (POP3), Simple Mail Transfer Protocol (SMTP) are the two major protocols used for accessing e-mail from servers.

g. What do you mean by checked exceptions ?

Ans. A checked exception is a type of exception that must be either caught or declared in the method in which it is thrown. For example, the java.io.IOException is a checked exception.

Section-B

2. Attempt any **three** of the following : **(7 × 3 = 21)**

a. Create an html page named as "Table.html" to display your class time table.

i. Provide the title as Time Table.

ii. Provide various colour options to the cells (Highlight the lab hours and elective hours with different colours).

Ans. **hyperlink.html :**

```
<html><body>
<a href="Pagelink.html">Different Tags</a><br>
<br><a href="table.html">Time Table</a><br>
</body></html>
```

Table.html :

```
    <html><head>
    <title>Time Table</title>
    </head><body>
    <B><center><h3>
    AKTU UNIVERSITY,<br>
    DEPARTMENT OF INFORMATION TECHNOLOGY<br>
    EFFECTIVE FROM : 04.02.19<br></center></b></h3>
    <table border="2" width="75%" cellspacing="2" cellpadding="25"
    align=center><h3>
    <caption align=left>Sem: III IT\CS
    & nbsp; & nbsp; & nbsp; & nbsp; & nbsp; &nbsp; & nbsp; &
    nbsp; & nbsp; & nbsp; & nbsp;  & nbsp; & nbsp; & nbsp; & nbsp;
    & nbsp; & nbsp; & nbsp; & nbsp; & nbsp; & nbsp; & nbsp; </
    caption></h3>
    <thead><tr>
    <th>Periods</th>
    <th ROWSPAN=2>1 <br>8.30-9.20</th>
    <th ROWSPAN=2>2<br>9.20-10.10</th>
    <th ROWSPAN=2>3<br>10.20-11.10</th>
    <th ROWSPAN=2>4<br>11.10-12.00</th>
    <th ROWSPAN=2>5<br>1.30-2.20</th>
    <th ROWSPAN=2>6<br>2.20-3.10</th>
    <th ROWSPAN=2>7<br>3.10-4.00</th>
</tr>
    <tr><th> Days</th>
    </thead>
    <tbody><tr>
    <th>Monday</th>
    <td>IT0302</td><td>IT0306</td><td>IT0304</td>
    IT0322 -Web Technology LAB
    <td>IT0306</td><td>IT0302</td>
    <td COLSPAN=2><center>IT0324</td></tr>
<tr>
    <th>Tuesday</th>
    <td COLSPAN =4 bgcolor="cyan"><center>IT0320/IT0322</
    td>
    <td>IT0310</td>
    <td>IT0304</td><td>IT308</td></tr>
<tr>
    <th>Wednesday</th>
```

```
        <td COLSPAN =2 bgcolor="red"><center>ELECTIVE - I</td>
        <td>IT0306</td><td>IT310</td>
        <td>IT304</td><td>IT308</td>
        <td>COUN</td></tr>
<tr>
        <th>Thursday</th>
        <td>IT302</td><td>IT0304</td>
        <td COLSPAN=2><center>PD0302</td>
        <td COLSPAN=3 bgcolor="cyan"><center>IT0320/IT0322</td>
</tr>
<tr>
        <th>Friday</th>
        <td>IT0308</td><td>IT0306</td>
        <td>IT0308</td><td>IT0302</td>
        <td COLSPAN=2 bgcolor="red"><center>ELECTIVE - I</td>
        <td></td>
        </tr></tbody></table></body>
```

b. Compare Java and JavaScript. Write a JavaScript program to define a user defined function for sorting the values in an array.

Ans. Comparison :

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.

Program to sort value using JavaScript :

```
<!DOCTYPE html>
<html><body>
<h2>JavaScript Array Sort</h2>
<p>The sort() method sorts an array alphabetically.</p>
<button onclick="myFunction()">Try it</button>
<p id="demo"></p>
```

```
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits;
function myFunction() {
fruits.sort();
document.getElementById("demo").innerHTML = fruits;
}
</script></body></html>
```

c. What is the difference between applet and application ? How is Java strongly associated with internet ? Draw a flowchart to show various Java tools are used in application development.

Ans. **Difference :**

S. No.	Applets	Application
1.	An applet is a java program which is embedded in web page to generate dynamic content.	An application is a java program that runs independently on client/server without web browser.
2.	The execution of the program does not start from main( ) method.	The execution of the program starts from the main( ) method.
3.	Applet cannot run program from local machine.	Application can run the program from local machine.
4.	It is used to perform small tasks.	It is used to perform large tasks.
5.	It can only access browser specific services.	It can access all kind of services available on the system.

1. Java is strongly associated with the internet because the first application program written in Java was HotJava, a browser to run the applet on internet.
2. So, the internet users use the java to create the applet programs and run them locally using a java-enabled browser's like HotJava.
3. The users can use the java-enabled browsers to download the applet located on the computer system anywhere in the internet and run it on their computer.

Flowchart to show various Java tools used in application development :

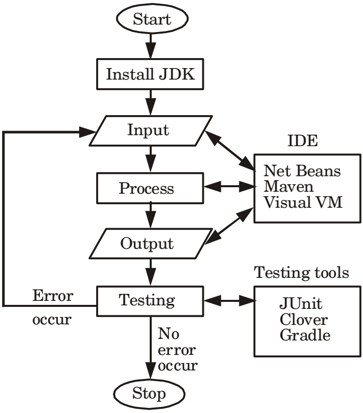


Fig. 1.

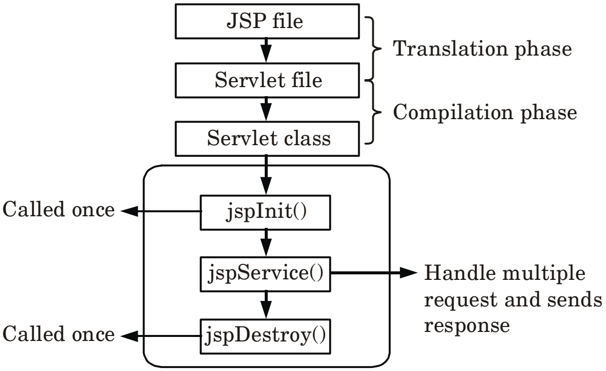
d. **Compare JSP and Servlet. Explain the life cycle of a JSP page with a suitable diagram. Also list any five action tags used in JSP.**

Ans. **Comparison :**

S. No.	JSP	Servlet
1.	JSP is protocol dependent.	Servlet is protocol independent.
2.	It can handle only HTTP and HTTPS protocol.	It can handle any type of protocol *i.e.*, FTP, HTTP.
3.	Time taken to generate response for first request is more.	Time taken to generate response for first request is less.
4.	JSP is a scripting language which can generate dynamic response.	Servlet is a Java program which can generate dynamic response.
5.	It is easy to code.	It is not easy to code.

Life cycle of JSP :

1. **Translation of JSP page to Servlet :** This is the first step of JSP life cycle. This translation phase deals with syntactic correctness of JSP. Here test.jsp file is translated to test.java.
2. **Compilation of JSP page :** Here the generated java servlet file (test.java) is compiled to a class file (test.class).
3. **Class loading :** Servlet class which has been loaded from JSP source is now loaded into container.



4. **Instantiation :** Here instance of the class is generated. The container manages one or more instance by providing response to requests.
5. **Initialization :** jspInit() method is called only once during the life cycle immediately after the generation of servlet instance from JSP.
6. **Request processing :** jspService() method is used to serve the raised requests by JSP. It takes request and response object as parameters. This method cannot be overridden.
7. **JSP cleanup :** In order to remove the JSP from use by the container or to destroy method for servlets jspDestroy() method is used. This method is called once, if we need to perform any cleanup task like closing open files, releasing database connections jspDestroy() can be overridden.

Five action tags used in JSP are :

1. <JSP : use Bean>
2. <JSP : set Property>
3. <JSP : include>
4. <JSP : attribute>
5. <JSP : body>

e. What are the uses of layout managers ? Give the name of classes that represents the layout managers. Explain any five layout managers.

Ans. Layout manager is used to place the component such as buttons, text boxes on the application.

Classes that represent the layout managers are :

1. BorderLayout
2. FlowLayout
3. GridLayout
4. CardLayout
5. GridBagLayout
6. BoxLayout

7. NoLayout
8. ScrollPaneLayout
9. SpringLayout

Five layout managers :

a. BorderLayout : This scheme lays out the component in five ways :

i. North-Northern part of the container

ii. South-Southern part of the container

iii. East-Eastern part of the container

iv. West-Western part of the container

v. Center-centered in the container

b. CardLayout : Allows for what Windows programmers have called "tabbed dialogs" or dynamic dialogs.

c. GridLayout : Allows for the layout of components in a grid-like fashion rather than "North" or "Center".

d. FlowLayout : Allows for component to be laid out in a row (or flow) and aligned (left, right, center).

e. None : No layout, the container will not attempt to reposition the components during an update.

Section-C

3. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. Explain the HTTP protocol. Mention three basic features of HTTP that make HTTP a simple but powerful protocol. Give its architecture.

Ans.

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. 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.

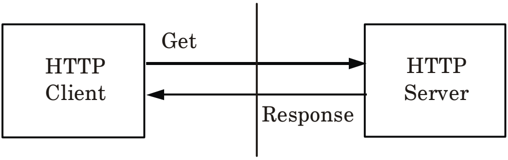
Features of HTTP protocol :

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.

Architecture :



HTTP Protocol

Fig. 2.

b. **What is XML ? Create a XML document of 10 students of final CSE. Add their roll number, marks obtained in 5 subjects, total marks and percentage. Save this XML document at the server, write a program that accepts student's roll number as input and returns the students marks, total percentage by taking student information for XML document.**

Ans. **XML :**

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.

Program :

XML document :

Student.XML

```
<?xml version = "1.0"?>
<!DOCTYPE STUDENTS SYSTEM "E:\XML1\STUDENT.dtd">
<STUDENTS><STUDENT><STUDENTDATA>
<Roll_No> 001</Roll_No>
<NAME> RAM</NAME>
<Marks>
<Marks 1> 70 </Marks 1>
<Marks 2> 80 </Marks 2>
<Marks 3> 50 </Marks 3>
<Marks 4> 60 </Marks 4>
<Marks 5> 70 </Marks 5>
<Total> 330 </Total>
<Percentage> 66.0 </Percentage>
</Marks>
</STUDENTDATA></STUDENT></STUDENTS>
```

NOTE : *Write the above code (bold part only) nine times again to enter the details of rest of the nine students.*

Student.dtd

```
<?xml version “1.0”?>
<!ELEMENT STUDENTS (STUDENT*)>
<!ELEMENT STUDENT (STUDENTDATA*)>
<!ELEMENT STUDENTDATA (Roll_No, Name, Marks, Total,
Percentage)>
<!ELEMENT Roll_No (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT Marks (#PCDATA)>
<!ELEMENT Total (#PCDATA)>
<!ELEMENT Percentage (#PCDATA)>
```

Student.html

```
<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0 Transitional/
/EN”
“E:\XML 1\STUDENT.dtd””>
<html>
<head COLOUR:RED><h1 style = “color : red”>
<MARQUEE DIRECTION = “RIGHT”><CENTER> COLLEGE OF
ENGINEERING AND TECHNOLOGY </CENTER> </
MARQUEE> </H1>
<title> STUDENT DETAILS DISPLAY <title>
</head>
<body style = “background-colour : PINK”> <H2 STYLE =
“COLOUR : BLUE”> <MARQUEE> <CENTER> DEPARTMENT
OF COMPUTER SCIENCE
</CENTER> </MARQUEE><BR></H2>
<MARQUEE DIRECTION = “DOWN”><H3 STYLE = “COLOR :
GREEN”><CENTER> FINAL CS STUDENTS DETAILS </
CENTER> </MARQUEE><BR><BR></H3><CENTER><TABLE
BORDER = “1”>
<THEAD><TR>
<TH> Roll_No </TH><TH> NAME </TH><TH> Marks 1 </TH>
<TH> Marks 2 </TH><TH> Marks 3 </TH><TH> Marks 4 </TH>
<TH> Marks 5 </TH><TH> Total </TH><TH>Percentage </TH>
</TR>
</THEAD>
<TFOOT><TR>
<TH COLSPAN = “4”> STUDENT CATALOG</TH>
</TFOOT>
<TR>
<TD> 001 </TD><TD> RAM </TD>
<TD> Marks 1 : 70 </TD><TD> Marks 2 : 80 </TD>
<TD> Marks 3 : 50 </TD><TD> Marks 4 : 60 </TD>
<TD> Marks 5 : 70 </TD><TD> 330 </TD><TD> 66.0 </TD>
</TR>
</TABLE></CENTER></body></html>
```

4. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. Discuss how frames play a big role in advertising on web. What roles do form play in making web page dynamic.

Ans. Role of frames in advertising on web :

1. A frame is a part of a web page or browser window which displays content independent of its container, with the ability to load content independently.
2. Frames play a big role in advertising on web by allowing the following advantages :
   a. Content can be loaded and navigated independently.
   b. Simple maintenance of content shared across all or most pages.
   c. Reducing the amount of bandwidth needed.
   d. Allowing several pieces of information to be viewed side by side.
3. All the above advantages helps in reducing the advertising budget and the audience can view the intended information easily.

Roles of form in making web page dynamic :

1. In hidden frame technique we could hide or minimise the size of frame.
2. The hidden frame is loaded with a web page that contains a form, and JavaScript is used to dynamically fill out the form making the web page dynamic.

b. What are exceptions and how they are handled ? Explain with an example. How we define a try and catch block ? Is it essential to catch all types of exceptions ?

Ans. 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.
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. 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.

b. The catch block can have one or more statements that are necessary to process the exception.

c. The catch statement is passed as a single parameter, which is reference to the type of exception object thrown by the try block.

d. 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 :

a. 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.

b. Finally block can be used to handle any exception generated within a try block.

c. It may be added immediately after the try block or after the last catch block as follows :

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

4. Throw :

a. Java supports "throw keyword" which is used if we want to throw our own exceptions.

b. We can do this by using the keyword throw as follows :

throw new Throwable_subclass;

No, it is not essential to catch all type of exceptions.

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.ArithmeticException: / by zero
finally block is always executed
rest of the code...
```

5. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. Create an html page named as "String_Math.html" and within the script tag define some string variables and use different string function to demonstrate the use of the predefined functions. Do the same for the math function.

Ans.

```
<html><body>
<script type="text/javascript">
var str1="W3Schools is great!"
document.write(str1.substr(2,6))
document.write("<br><br>")
document.write(str1.substring(2,6))
// string length
```

```
var str2="Web Enabling Tools is Cool!"
document.write("<p>" + str2 + "</p>")
document.write("str.length")
// Square root of a number
var property_value = Math.SQRT1_2
document.write("Property Value is : " + property_value);
var value = Math.exp(1);
document.write("First Test Value : " + value );
var value = Math.exp(30);
document.write("<br />Second Test Value : " + value );
var value = Math.exp(– 1);
document.write("<br />Third Test Value : " + value );
// Find maximum of a number
var value = Math.max(10, 20, – 1, 100);
document.write("First Test Value : " + value );
var value = Math.max(– 1, – 3, – 40);
document.write("<br />Second Test Value : " + value );
</script>
```

b. What are the advantages and drawback of applet ? Write a Java program to create an applet for calculator and also perform event handling on each button.

Ans. **Advantages of applets :**

1. Applets are platform independent.
2. Applets are quite secure and safe to use.
3. Applets cache quickly.
4. Applet increase interactivity for users.
5. Database integration is another important advantage of applets.

Drawbacks of applets :

1. Applets do not access client-side resources, like such as file, operating system.
2. Applet cannot work with native methods.
3. Applet can only extract information about client-machine *i.e.*, its name, Java version, OS, version etc.
4. Mobile browsers which are running on IOS or Android do not support applets.

Program :

```
import java.applet.*;
import java.awt.*;
import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;
import java.awt.event.TextEvent;
import java.awt.event.TextListener;;
public class calculator extends Applet implements
ActionListener, TextListener {
String s,s1,s2,s3,s4;
Button b1,b2,b3,b4,b5,b6,b7,b8,b9,b0;
Button add,sub,eq,cl,mul,div;
TextField t1;
int a,b,c;
public void init() {
t1=new TextField(10);
b1=new Button(“1”);  b2=new Button(“2”); b3=new Button(“3”);
b4=new Button(“4”); b5=new Button(“5”); b6=new Button(“6”);
b7=new Button(“7”);  b8=new Button(“8”); b9=new Button(“9”);
b0=new Button(“0”);
add=new Button(“+”); sub=new Button(“–”);
mul=new Button(“*”); div=new Button(“/”);
eq=new Button(“=”); cl=new Button(“Clear”);
GridLayout gb=new GridLayout(4,5);
setLayout(gb);
add(t1);add(b1);add(b2); add(b3);add(b4);add(b5);
add(b6); add(b7);add(b8); add(b9);add(b0);add(add);
add(sub);add(mul);add(div); add(eq);add(cl);
b1.addActionListener(this); b2.addActionListener(this);
b3.addActionListener(this); b4.addActionListener(this);
b5.addActionListener(this); b6.addActionListener(this);
b7.addActionListener(this); b8.addActionListener(this);
b9.addActionListener(this); b0.addActionListener(this);
add.addActionListener(this); sub.addActionListener(this);
mul.addActionListener(this); div.addActionListener(this);
eq.addActionListener(this); cl.addActionListener(this);
paint();
//t1.addTextListener(this); }
public void paint() {
setBackground(Color.green); }
public void actionPerformed(ActionEvent e) {
s=e.getActionCommand();
if(s.equals(“0”) || s.equals(“1”) || s.equals(“2”) ||
s.equals(“3”) || s.equals(“4”) || s.equals(“5”) || s.equals(“6”) ||
s.equals(“7”) || s.equals(“8”) ||
s.equals(“9”) || s.equals(“0”)) {
s1=t1.getText()+s;
t1.setText(s1); }
if(s.equals(“+”)) {
s2=t1.getText();
t1.setText(“”);
s3=“+”; }
if(s.equals(“–”)) {
s2=t1.getText();
t1.setText(“”);
s3=“–”; }
if(s.equals(“*”)) {
s2=t1.getText();
t1.setText("");
s3="*"; }
if(s.equals("–")) {
s2=t1.getText();
t1.setText("");
s3="–"; }
if(s.equals("=")) {
s4=t1.getText();
a=Integer.parseInt(s2);
b=Integer.parseInt(s4);
if(s3.equals("+"))
c=a+b;
if(s3.equals("–"))
c=a–b;
t1.setText(String.valueOf(c)); }
if(s.equals("Clear")) {
t1.setText(""); } }
public void textValueChanged(TextEvent e) { }}
```

6. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. What do you mean by CSS ? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system. Mention how can you integrate CSS on a web page?

Ans. **CSS :**

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>.

Program :

```
    <HTML><HEAD><STYLE>
    H1 {colour: red; font-family: arial; font-size: 2.50 em}
</STYLE></HEAD><BODY>
<H1> This is the H1 element </H1>
</BODY></HTML>
```

Integration of CSS on a web page :

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>
```

b. What is difference between session and cookies ? Write a servlet program for servlet login and logout using cookies.

Ans. **Difference :**

S. No.	Session	Cookies
1.	Sessions are stored in server.	Cookies are stored in the user's browser.
2.	A session is available as long as the browser is opened. User can not disable the session.	Cookies can keep information in the user's browser until deleted by user or set as per the timer.
3.	It will be destroyed if we close the browser.	It will not be destroyed even if we close the browser.
4.	It can store any object.	Cookies can only store string.
5.	Session cannot be same for future reference.	We can save cookies for future reference.

Program :

Index.html :

```
<!DOCTYPE html><html><head>
<meta charset="ISO-8859-1">
<title>Servlet Login Example</title>
</head><body>
<h1>Welcome to Login App by Cookie</h1>
<a href="login.html">Login</a>|
<a href="LogoutServlet">Logout</a>|
<a href="ProfileServlet">Profile</a></body></html>
```

link.html :

```
<a href="login.html">Login</a>  |
<a href="LogoutServlet">Logout</a>  |
<a href="ProfileServlet">Profile</a>
<hr>
```

login.html :

```
<form action="LoginServlet" method="post">
Name:<input type="text" name="name"><br>
Password:<input type="password" name="password"><br>
<input type="submit" value="login"></form>
```

LoginServlet.java :

```
package com.javatpoint;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
```

```
public class LoginServlet extends HttpServlet {
protected void doPost(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out=response.getWriter();
request.getRequestDispatcher("link.html").include(request,
response);
String name=request.getParameter("name");
String password=request.getParameter("password");
if(password.equals("admin123")){
out.print("You are successfully logged in!");
out.print("<br>Welcome, "+name);
Cookie ck=new Cookie("name",name);
response.addCookie(ck);
}else{
out.print("sorry, username or password error!");
request.getRequestDispatcher("login.html").include(request,
response); }
out.close(); } }
```

LogoutServlet.java :

```
package com.javatpoint;
import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.Cookie;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;
public class LogoutServlet extends HttpServlet {
protected void doGet(HttpServletRequest request,
HttpServletResponse response)
throws ServletException, IOException {
response.setContentType("text/html");
PrintWriter out=response.getWriter();
request.getRequestDispatcher("link.html").include(request,
response);
Cookie ck=new Cookie("name","");
ck.setMaxAge(0);
response.addCookie(ck);
out.print("you are successfully logged out!"); } }
```

7. Attempt any **one** of the following questions : **(7 × 1 = 7)**

a. What is JDBC ? Explain the drivers used in JDBC. Write a JDBC program for insert and display the record of employees using prepared statement.

Ans. **JDBC :**

1. JDBC (Java Database Connectivity) is a Java API that manages connection to database, issuing queries and commands and handling result sets obtained from the database.
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.

Drivers in JDBC :

1. **JDBC-ODBC bridge driver (Type 1 driver) :**
   a. These drivers are the bridge drivers such as JDBC-ODBC bridge.
   b. These drivers rely on an intermediary such as ODBC to transfer the SQL calls to the database.
   c. Bridge drivers often rely on native code, although the JDBC-ODBC library native code is part of the Java-2 virtual machine.

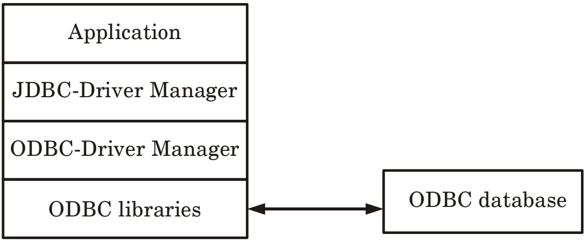


Fig. 3. JDBC-ODBC bridge driver.

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

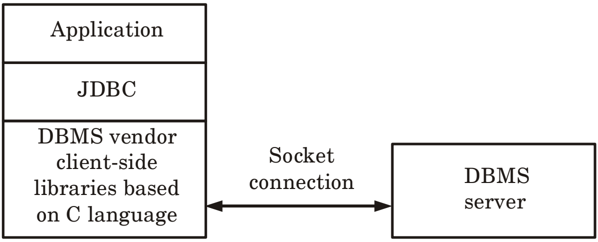


Fig. 4. Native API partly Java driver.

3. **JDBC net pure Java driver (Type 3 driver) :**
   a. JDBC net pure Java driver consists of JDBC and DBMS independent protocol driver.
   b. Here the calls are translated and sent to middle tier server through the socket.
   c. The middle tier contacts the database.
   d. Type 3 drivers call the database API on the server.
4. **Native protocol pure Java driver (Type 4 driver) :**
   a. A native protocol Java driver contains JDBC calls that are converted directly to the network protocol used by the DBMS server.
   b. This driver interacts directly with database.

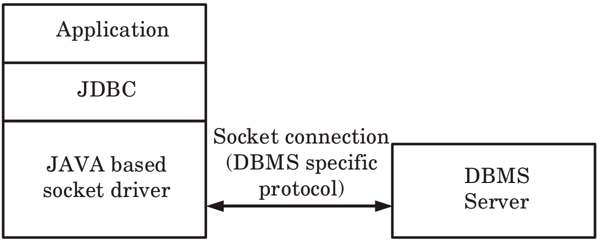


Fig. 5. Native protocol pure Java driver.

   c. It does not require any native database library. So, it is also called thin driver.

Program :

```
import java.sql.*; import java.io.*;
public class PreparedStatementDemo1 {
Connection con;
PreparedStatement ps;
public PreparedStatementDemo1() {
try {
Class.forName("com.mysql.jdbc.Driver");
con = DriverManager.getConnection("jdbc:mysql://localhost/
test?user=root&password=root");
} catch (Exception e) { e.printStackTrace(); } }
// add customer detail
public String addCustomer(String custid, String name, String
address,
String contact) {
String status = "";
try {
ps = con.prepareStatement("insert into Customer values(?,?,?,?)");
ps.setString(1, custid); ps.setString(2, name);
ps.setString(3, address); ps.setString(4, contact);
int i = ps.executeUpdate();
if (i != 0) {
```

```
status = “Inserted”;
} else {
status = “Not Inserted”; }
} catch (Exception e) { e.printStackTrace(); }
return status; }
// customer record
public void searchCustomer(String custid) {
String sql = “”;
if (custid.trim().length() == 0) {
sql = “select * from Customer”;
} else {
sql = “select * from Customer where custid=” + custid + “”; }
try {
ps = con.prepareStatement(sql);
ResultSet res = ps.executeQuery();
while (res.next()) {
System.out.print(res.getString(1));
System.out.print(res.getString(2));
System.out.print(res.getString(3));
System.out.println(res.getString(4)); }
} catch (SQLException e) {e.printStackTrace(); } }
public String deleteCustomer(String custId) {
String status = “”;
try {
ps = con.prepareStatement(“delete from Customer where
custid=?”);
ps.setString(1, custId);
int i = ps.executeUpdate();
if (i != 0) {
status = “Customer details deleted”;
} else {
status = “Customer details not deleted”; }
} catch (Exception e) {e.printStackTrace(); }
return status; }
public void menuDisplay() {
try {
BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
int ch = 0;
while (true) {
System.out.println(“== Customer Management System == \n”
+ “1. Add Customer \n” + “2. Display Customer’s record \n”
+ “3. Exit \n” + “Enter Choice \n”);
String str1 = br.readLine().toString();
ch = Integer.parseInt(str1);
switch (ch) {
case 1: {
```

```
System.out.println("Enter Customer Id");
String custId = br.readLine();
System.out.println("Enter Customer Name");
String custName = br.readLine();
System.out.println("Enter  Customer Address");
String custAddress = br.readLine();
System.out.println("Enter Customer Contact No.");
String custContact = br.readLine();
System.out.println(addCustomer(custId, custName,
custAddress, custContact));
break; }
case : 2{
System.out.println("Enter Customer Code to display record");
String custId = br.readLine();
searchCustomer(custId);
break; }
case 3: {
System.exit(0); }
default:
break; } }
} catch (Exception e) {
e.printStackTrace(); } }
public static void main(String[] args) {
PreparedStatementDemo1 obj = new
PreparedStatementDemo1();
obj.menuDisplay(); } }
```

b. What are XML parsers ? Explain the types of parsers with their advantages and disadvantages.

Ans. **XML parser :**

1. An XML parser is a software library or package that provides interfaces for client applications to work with an XML document.
2. The XML Parser is designed to read the XML and create a way for programs to use XML.
3. XML parser validates the document and check that the document is well formatted.

There are two types of XML Parsers :

1. **DOM :**
   a. The Document Object Model (DOM) is a platform-independent and language-independent standard object model for representing HTML or XML and related formats.
   b. The DOM defines a standard for accessing documents like XML and HTML.

   Advantages :

   i. It supports both read and write operations and the API is very simple to use.

ii. It is preferred when random access to widely separated parts of a document is required.

Disadvantages :

i. It is memory inefficient.

ii. It is comparatively slower than other parsers.

2. SAX :

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

b. SAX is an event-driven API that allows us to interpret a web file that uses XML.

c. SAX takes the control of event specifies by the programmer and handles the situation.

Advantages :

i. It is simple and memory efficient.

ii. It is very fast and works for huge documents.

Disadvantages :

i. It is event based so its API is less intuitive.

ii. Clients never know the full information because the data is broken into pieces.

☺☺☺

B. Tech.

(SEM. V) ODD SEMESTER THEORY EXAMINATION, 2019-20

WEB TECHNOLOGIES

Time : 3 Hours **Max. Marks : 70**

Note : Attempt **all** sections. If require any missing data; then choose suitably.

Section-A

1. Attempt **all** questions in brief. **(2 × 7 = 14)**

a. **Define Path in JDK. How path is different from class path ?**

b. **Define webpage with its type. Discuss responsive webpage with example.**

c. **Define box model in CSS with block diagram.**

d. **Describe garbage collection and demonstrate how it is functioning ?**

e. **Define DHTML with suitable example.**

f. **Compare JDBC and ODBC. List the different types of JDBC drivers.**

g. **Discuss about tomcat server. How to set the Class path for servlet in tomcat server ?**

Section-B

2. Attempt any **three** of the following : **(7 × 3 = 21)**

a. **Compare object-oriented programming and object-based programming with example. List the features of object-oriented programming. Write a program in Java to demonstrate use of this keyword in constructor.**

b. **Define thread. How to create a thread in java ? Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 5.**

c. **Discuss XML. Which technology is used to define the structure of XML document ? Explain and demonstrate with an example.**

d. **Discuss socket and server socket in Java with its package. Write a program in Java to demonstrate, how the communication is establish between client and server ?**

e. **Discuss EJB. Explain EJB architecture. What are its various types ?**

Section-C

3. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. **Explain AWT and its controls. How the layout manager manage the AWT controls ? Write a program to demonstrate graphics (*i.e.* line, circle, rectangle etc.) using Frame, Panel, and layout manager.**

b. **Explain Applet with its life cycle. Write a program to demonstrate simple java applet to display any image. Compare Applets over HTML.**

4. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. **Discuss DTD. How the DTD is different from XSD ? Demonstrate to create a XML document of 10 students of third year. Add their roll numbers, marks obtained in 5 subjects, total marks and percentage and validate using DTD.**

b. **Explain CSS. What are the CSS frameworks? Explain in brief. What are the different ways of using the stylesheet ? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system.**

5. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. **Discuss AJAX. Explain the application of AJAX with the help of suitable examples.**

b. **Compare Java and JavaScript. Explain and demonstrate 5 different types of objects in JavaScript with example.**

6. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. **Explain JDBC application architecture. List the various types of JDBC drivers. Discuss the steps to connect database with the web application using JDBC. Write a program to demonstrate how to retrieve the data from a table using JDBC API.**

b. **Explain JavaBeans. Why they are used? Discuss setter and getter method with Java code.**

7. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. **Explain servlets with its life cycle. How its life cycle is different from the life cycle of JSP ? Explain with an example.**

b. **Discuss JSP in details. What are JSP directives ? Explain various types of directives with suitable example.**

☺☺☺

SOLUTION OF PAPER (2019-20)

Note : Attempt **all** sections. If require any missing data; then choose suitably.

Section-A

1. Attempt **all** questions in brief. **(2 × 7 = 14)**

a. Define Path in JDK. How path is different from class path ?

Ans. Path variable is set for providing path for all Java tools like Java, Javac, Javap, Javah, jar, Appletviewer which are used in Java programming. All these tools are available in bin folders so we set path upto bin folders.

Difference :

S. No.	Path	Class path
1.	Path variable is set for providing path for all java tools like Java, Javac, Javap, Javah, jar, Appletviewer	Class path variable is set for providing path of all Java classes which is used in our application.
2.	It contains a path to the Java tools.	It contains a path of the classes provided by JDK.

b. Define webpage with its type. Discuss responsive webpage with example.

Ans. A webpage is a document commonly written in HTML that is accessible through the Internet or other networks using an Internet browser.

Types of webpage :

1. **Dynamic webpages :** Dynamic webpages are behavioral and have the capacity to produce distinctive content for different visitors.
2. **Static webpages :** Static webpages will remain same for the time until and unless someone changes it manually.

Responsive webpage :

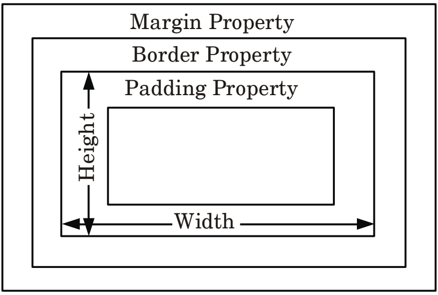
1. A responsive webpage is one that has been designed to respond, or adapt, based on the technology and type of computing device used by the visitor to display the site.
2. It is basically one website design that will look good at any size from a large desktop LCD monitor to the smaller screens we use on smart phones and tablets.
3. Facebook, Youtube are the example of responsive webpage.

c. Define box model in CSS with block diagram.

Ans. CSS box model is a container which contains multiple properties including borders, margin, padding and the content itself. It is used to create the design and layout of webpages. It can be used as a toolkit for customizing the layout of different elements.

Block diagram :

CSS Box-Model Property



d. Describe garbage collection and demonstrate how it is functioning ?

Ans. Garbage Collection (GC) is a form of automatic memory management. The garbage collector attempts to reclaim garbage, or memory occupied by objects that are no longer in use by the program.

Functioning of garbage collection :

1. In the first step, unreferenced objects are identified and marked as ready for garbage collection.
2. In the second step, marked objects are deleted.
3. Memory can be compacted after the garbage collector deletes objects, so remaining objects are in a contiguous block at the start of the heap.
4. The compaction process makes it easier to allocate memory to new objects sequentially after the block of memory allocated to existing objects.

e. Define DHTML with suitable example.

Ans. **DHTML :** DHTML is a collection of technologies used together to create interactive and dynamic websites.

Example :

```
<!DOCTYPE html>
<html lang="en-US">
<head>
<title>Welcome to Quantum Page </title>
</head>
<body>
<h1 onclick="this.style.color='red'">Hi Everyone!</h1>
```

```
</body>
</html>
```

f. Compare JDBC and ODBC. List the different types of JDBC drivers.

Ans. Comparison :

S. No.	JDBC	ODBC
1.	JDBC is language and platform dependent.	ODBC is language and platform independent.
2.	JDBC is Java Database Connectivity	ODBC is Open Database Connectivity.
3.	Code is easy to understand.	Code is complex.

Different JDBC drivers :

1. JDBC-ODBC bridge driver
2. Native API driver
3. Network protocol driver
4. JDBC Net driver

g. Discuss about tomcat server. How to set the Class path for servlet in tomcat server ?

Ans. Tomcat server :

Tomcat server is an application server that executes Java servlets and renders web pages that include Java Server Page coding. Components of Tomcat server are :

1. Catalina
2. Coyote
3. Jasper

To set class path for servlet in Tomcat server :

1. Copy servlet-api.jar file location and set the class path in environment variable.
2. If our class path is already set for core java programming, we need to edit class path variable. For edit classpath just put ';' at end of previous variable and paste new copied location (without deleting previous classpath variable).

Section-B

2. Attempt any **three** of the following : **(7 × 3 = 21)**

a. Compare object-oriented programming and object-based programming with example. List the features of object-oriented programming. Write a program in Java to demonstrate use of this keyword in constructor.

Ans. **Comparison :**

S. No.	Object-Oriented Programming	Object-Based Programming
1.	It supports all the features of OOPS.	It supports the usage of object and encapsulation.
2.	It also supports inheritance and polymorphism.	It does not support inheritance or, polymorphism or, both.
3.	It supports built-in objects.	It does not supports built-in objects.
4.	C#, Java, VB.Net are the examples of object-oriented languages.	JavaScript, VB are the examples of object-based languages.

Features of Object-Oriented programming are :

1. Inheritance
2. Polymorphism
3. Data hiding
4. Encapsulation
5. Overloading
6. Reusability

“this” keyword with constructor :

1. “this” keyword can be used inside the constructor to call another overloaded constructor in the same class. It is called the explicit constructor invocation.
2. This occurs if a class has two overloaded constructors, one without argument and another with the argument. Then “this” keyword can be used to call the constructor with an argument from the constructor without argument. This is required as the constructor cannot be called explicitly.

Program :

```
class JBT {
JBT() {
this(“JBT”);
System.out.println(“Inside Constructor without parameter”);
}
JBT(String str) {
System.out.println(“Inside Constructor with String parameter as
” + str);
}
public static void main(String[ ] args) {
JBT obj = new JBT();
}
}
```

Output :

```
Inside Constructor with String parameter as JBT
Inside Constructor without parameter
```

b. **Define thread. How to create a thread in java ? Write a program that executes two threads. One thread will print the even numbers and another thread will print odd numbers from 1 to 5.**

Ans. **Thread :** A thread is a light weighted process which runs concurrently with other threads. All threads of program define a separate path of execution.

Create a thread in Java :

```
public class Thread Test
{
public static void main (string [ ] args)
{
System.out.println (“Constructing the thread....”);
BytePrinter bp = new BytePrinter ( );
System.out.println (“Starting the thread....”);
bp start ( );
System.out.println (“The thread has been started”);
System.out.println (“The main ( ) method is finishing”);
return;
}
}
```

Program :

1. In the first step, we will implement the Runnable interface to define the logic of both threads. In the run method, we check if the number is even or odd.
2. If the number is even, we call the printEven method of the Printer class, else we call the printOdd method :

```
class TaskEvenOdd implements Runnable {
private int max;
private Printer print;
private boolean isEvenNumber;
// standard constructors
@Override
public void run() {
int number = isEvenNumber ? 2 : 1;
while (number <= max) {
if (isEvenNumber) {
print.printEven(number);
} else {
print.printOdd(number);
}
number += 2;
}
}
}
```

3. We define the Printer class as :

```
class Printer {
private volatile boolean isOdd;
synchronized void printEven(int number) {
while (!isOdd) {
try {
wait();
} catch (InterruptedException e) {
Thread.currentThread().interrupt();
}
}
System.out.println(Thread.currentThread().getName() + ":" +
number);
isOdd = false;
notify();
}
synchronized void printOdd(int number) {
while (isOdd) {
try {
wait();
} catch (InterruptedException e) {
Thread.currentThread().interrupt();
}
}
System.out.println(Thread.currentThread().getName() + ":" +
number);
isOdd = true;
notify();
}
}
```

4. In the main method, we use the defined class to create two threads. We create an object of the Printer class and pass it as the parameter to the TaskEvenOdd constructor:

```
public static void main(String... args) {
Printer print = new Printer();
Thread t1 = new Thread(new TaskEvenOdd(print, 5, false),"Odd");
Thread t2 = new Thread(new TaskEvenOdd(print, 5, true),"Even");
t1.start();
t2.start();
}
```

c. Discuss XML. Which technology is used to define the structure of XML document ? Explain and demonstrate with an example.

Ans. **XML :**

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.

Program :

XML document :

Student.XML

```
<?xml version = “1.0”?>
<!DOCTYPE STUDENTS SYSTEM “E:\XML1\STUDENT.dtd”>
<STUDENTS><STUDENT><STUDENTDATA>
<Roll_No> 001</Roll_No>
<NAME> RAM</NAME>
<Marks>
<Marks 1> 70 </Marks 1>
<Marks 2> 80 </Marks 2>
<Marks 3> 50 </Marks 3>
<Marks 4> 60 </Marks 4>
<Marks 5> 70 </Marks 5>
<Total> 330 </Total>
<Percentage> 66.0 </Percentage>
</Marks>
</STUDENTDATA></STUDENT></STUDENTS>
```

NOTE : *Write the above code (bold part only) nine times again to enter the details of rest of the nine students.*

Student.dtd

```
<?xml version “1.0”?>
<!ELEMENT STUDENTS (STUDENT*)>
<!ELEMENT STUDENT (STUDENTDATA*)>
<!ELEMENT STUDENTDATA (Roll_No, Name, Marks, Total,
Percentage)>
<!ELEMENT Roll_No (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT Marks (#PCDATA)>
<!ELEMENT Total (#PCDATA)>
<!ELEMENT Percentage (#PCDATA)>
```

Student.html

```
<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0
Transitional//EN”
“E:\XML 1\STUDENT.dtd””>
<html>
```

```
<head COLOUR:RED><h1 style = “color : red”>
<MARQUEE DIRECTION = “RIGHT”><CENTER> COLLEGE
OF ENGINEERING AND TECHNOLOGY </CENTER> </
MARQUEE> </H1>
<title> STUDENT DETAILS DISPLAY <title>
</head>
<body style = “background-colour : PINK”> <H2 STYLE =
“COLOUR : BLUE”> <MARQUEE> <CENTER> DEPARTMENT
OF COMPUTER SCIENCE
</CENTER> </MARQUEE><BR></H2>
<MARQUEE DIRECTION = “DOWN”><H3 STYLE = “COLOR :
GREEN”><CENTER> FINAL CS STUDENTS DETAILS </
CENTER> </MARQUEE><BR><BR></H3><CENTER><TABLE
BORDER = “1”>
<THEAD><TR>
<TH> Roll_No </TH><TH> NAME </TH><TH> Marks 1 </TH>
<TH> Marks 2 </TH><TH> Marks 3 </TH><TH> Marks 4 </TH>
<TH> Marks 5 </TH><TH> Total </TH><TH>Percentage </TH>
</TR>
</THEAD>
<TFOOT><TR>
<TH COLSPAN = “4”> STUDENT CATALOG</TH>
</TFOOT>
<TR>
<TD> 001 </TD><TD> RAM </TD>
<TD> Marks 1 : 70 </TD><TD> Marks 2 : 80 </TD>
<TD> Marks 3 : 50 </TD><TD> Marks 4 : 60 </TD>
<TD> Marks 5 : 70 </TD><TD> 330 </TD><TD> 66.0 </TD>
</TR>
</TABLE></CENTER></body></html>
```

DTD is use to define the structure of XML document.

d. Discuss socket and server socket in Java with its package. Write a program in Java to demonstrate, how the communication is establish between client and server ?

Ans. Socket :

1. TCP/IP client sockets are used to implement bi-directional, point-to-point, stream-based connections between hosts on the Internet.
2. 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.
3. The creation of a Socket object implicitly establishes a connection between the client and server.
4. Following are the two constructors used to create client socket :
   a. **Socket(String hostName, int port) :** Creates a socket connecting the local host to the named host, port and can throw an Unknown HostException or an IOException.

b. **Socket(InetAddress ipAddress, int port) :** Creates a socket using a pre-existing InetAddress object, a port and can throw an IOException.

5. Following methods are used by TCP/IP client socket :
   a. **InetAddress getInetAddress() :** Returns the InetAddress associated with the Socket object.
   b. **Int getPort() :** Returns the remote port to which the Socket object is connected.
   c. **Int getLocalPort() :** Returns the local port to which the Socket object is connected.

Server socket :

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().

For example :

```
// A Java program for a Client
import java.net.*;
import java.io.*;
public class Client
{
// initialize socket and input output streams
private Socket socket = null;
private DataInputStream input = null;
private DataOutputStream out = null;
// constructor to put ip address and port
public Client(String address, int port)
{
// establish a connection
try
{
socket = new Socket(address, port);
System.out.println(“Connected”);
// takes input from terminal
input = new DataInputStream(System.in);
// sends output to the socket
out = new DataOutputStream(socket.getOutputStream());
}
catch(UnknownHostException u)
{
System.out.println(u);
}
catch(IOException i)
{
System.out.println(i);
}
// string to read message from input
String line = “”;
// keep reading until “Over” is input
while (!line.equals(“Over”))
{
try
{
line = input.readLine();
out.writeUTF(line);
}
catch(IOException i)
{
System.out.println(i);
}
}
// close the connection
try
{
input.close();
out.close();
socket.close();
}
catch(IOException i)
{
System.out.println(i);
}
}
public static void main(String args[ ])
{
Client client = new Client(“127.0.0.1”, 5000);
}
}
```

e. Discuss EJB. Explain EJB architecture. What are its various types ?

Ans. **EJB :**

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

EJB architecture :

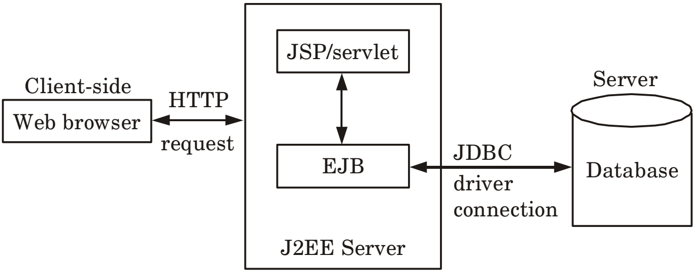


Fig. 1. Architecture.

The EJB architecture is an extension of web architecture.

Working of EJB architecture :

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

Types of EJB :

1. **Entity bean :** Entity beans represent persistent data storage. Entity beans are used for modeling the business concept.
2. **Session bean :** Session beans are used for managing processes or tasks. Hence, session beans are used for managing activities.
3. **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.

Section-C

3. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. Explain AWT and its controls. How the layout manager manage the AWT controls ? Write a program to demonstrate graphics (*i.e.* line, circle, rectangle etc.) using Frame, Panel, and layout manager.

Ans. AWT :

1. The AWT stands for Abstract Window Toolkit.
2. AWT is a library of class which provides GUI tools to develop GUI application and applet.
3. It provides many classes for programmers to use. It is the connection between our application and the native GUI.
4. 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 :
   - **a. Containers class :** Frame, Dialog, Panel, Applet etc.
   - **b. Components class :** TextField, Button, Checkbox, Scrollbar, Label, List etc.
   - **c. Custom graphics class :** Colour, Font, Dimensions etc.

The AWT supports following types of controls :

i. Buttons
ii. Checkbox
iii. CheckboxGroup
iv. Choice
v. Label
vi. List
vii. Scrollbar
viii. TextField
ix. TextArea

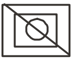
LayoutManager :

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

Program :

```
package com.mkyong;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Graphics2D;
import java.awt.Rectangle;
import java.awt.Shape;
import java.awt.geom.Ellipse2D;
import java.awt.geom.Line2D;
import java.awt.geom.RoundRectangle2D;
import javax.swing.JFrame;
import javax.swing.JPanel;
import javax.swing.SwingUtilities;
public class DrawShapes extends JFrame {
private static final long serialVersionUID = 1L;
public DrawShapes() {
setSize(new Dimension(320, 320));
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
setVisible(true);
JPanel p = new JPanel() {
@Override
public void paintComponent(Graphics g) {
Graphics2D g2 = (Graphics2D) g;
Shape line = new Line2D.Double(3, 3, 303, 303);
Shape rect = new Rectangle(3, 3, 303, 303);
Shape circle = new Ellipse2D.Double(100, 100, 100, 100);
Shape roundRect = new RoundRectangle2D.Double(20, 20, 250,
250, 5, 25);
g2.draw(line);
g2.draw(rect);
g2.draw(circle);
g2.draw(roundRect);
}
};
setTitle("My Shapes");
this.getContentPane().add(p);
}
public static void main(String arg[ ]) {
SwingUtilities.invokeLater(new Runnable() {
@Override
public void run() {
// TODO Auto-generated method stub
new DrawShapes();
}
});
}
}
```

Output :



b. Explain Applet with its life cycle. Write a program to demonstrate simple java applet to display any image. Compare Applets over HTML.

Ans. **Applet :**

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. Applets are used to make the website more dynamic and entertaining.
4. All applets are sub-classes (either directly or indirectly) of java.applet.Applet class.
5. 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.

d. destroy( ) : This method is only called when the browser shuts down normally.

e. paint( ) : Invoked immediately after the start( ) method, and also any time the applet needs to repaint itself in the browser.

Program :

```
import java.awt.*;
import java.applet.*;
public class DisplayImage extends Applet {
Image picture;
public void init() {
picture = getImage(getDocumentBase(),“sonoo.jpg”);
}
public void paint(Graphics g) {
g.drawImage(picture, 30,30, this);
}
}
```

Comparison :

S. No.	Applet	HTML
1.	Applet is a JAVA program.	HTML is Hyper Text Markup Language.
2.	It is embedded in the HTML code to provide functionality to the webpage.	It is used to provide position of the text in the webpage.
3.	It need Java package to run on JVM.	HTML do not need any package to run on browsers.

4. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. Discuss DTD. How the DTD is different from XSD ? Demonstrate to create a XML document of 10 students of

third year. Add their roll numbers, marks obtained in 5 subjects, total marks and percentage and validate using DTD.

Ans. **DTD :**

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.

Difference :

S. No.	DTD	XSD
1.	DTD is a set of markup declarations that define a document type for an SGML.	XSD specifies how to describe the elements in an XML document formally.
2.	DTD stands for Document Type Definition.	XSD stands for XML Schema Definition.
3.	DTD provides less control over the XML structure.	XSD provides more control over the XML structure.
4.	DTD does not support data types.	XSD supports data types.

Program :

XML document :

Student.XML

```
<?xml version = “1.0”?>
<!DOCTYPE STUDENTS SYSTEM “E:\XML1\STUDENT.dtd”>
<STUDENTS><STUDENT><STUDENTDATA>
<Roll_No> 001</Roll_No>
<NAME> RAM</NAME>
<Marks>
<Marks 1> 70 </Marks 1>
<Marks 2> 80 </Marks 2>
<Marks 3> 50 </Marks 3>
<Marks 4> 60 </Marks 4>
<Marks 5> 70 </Marks 5>
<Total> 330 </Total>
<Percentage> 66.0 </Percentage>
</Marks>
</STUDENTDATA></STUDENT></STUDENTS>
```

NOTE : *Write the above code (bold part only) nine times again to enter the details of rest of the nine students.*

Student.dtd

```
<?xml version “1.0”?>
<!ELEMENT STUDENTS (STUDENT*)>
<!ELEMENT STUDENT (STUDENTDATA*)>
<!ELEMENT STUDENTDATA (Roll_No, Name, Marks, Total,
Percentage)>
<!ELEMENT Roll_No (#PCDATA)>
<!ELEMENT NAME (#PCDATA)>
<!ELEMENT Marks (#PCDATA)>
<!ELEMENT Total (#PCDATA)>
<!ELEMENT Percentage (#PCDATA)>
```

Student.html

```
<!DOCTYPE html PUBLIC “-//W3C//DTD XHTML 1.0
Transitional//EN”
“E:\XML 1\STUDENT.dtd””>
<html>
<head COLOUR:RED><h1 style = “color : red”>
<MARQUEE DIRECTION = “RIGHT”><CENTER> COLLEGE
OF ENGINEERING AND TECHNOLOGY </CENTER> </
MARQUEE></H1>
<title>STUDENT DETAILS DISPLAY <title>
</head>
<body style = “background-colour : PINK”> <H2 STYLE =
“COLOUR : BLUE”> <MARQUEE> <CENTER> DEPARTMENT
OF COMPUTER SCIENCE
</CENTER> </MARQUEE><BR></H2>
<MARQUEE DIRECTION = “DOWN”><H3 STYLE = “COLOR :
GREEN”><CENTER> FINAL CS STUDENTS DETAILS </
CENTER> </MARQUEE><BR><BR></H3><CENTER><TABLE
BORDER = “1”>
<THEAD><TR>
<TH> Roll_No </TH><TH> NAME </TH><TH> Marks 1 </TH>
<TH> Marks 2 </TH><TH> Marks 3 </TH><TH> Marks 4 </TH>
<TH> Marks 5 </TH><TH> Total </TH><TH>Percentage </TH>
</TR>
</THEAD>
<TFOOT><TR>
<TH COLSPAN = “4”> STUDENT CATALOG</TH>
</TFOOT>
<TR>
<TD> 001 </TD><TD> RAM </TD>
<TD> Marks 1 : 70 </TD><TD> Marks 2 : 80 </TD>
<TD> Marks 3 : 50 </TD><TD> Marks 4 : 60 </TD>
<TD> Marks 5 : 70 </TD><TD> 330 </TD><TD> 66.0 </TD>
</TR>
</TABLE></CENTER></body></html>
```

b. Explain CSS. What are the CSS frameworks? Explain in brief. What are the different ways of using the stylesheet ? Write a CSS rule that makes all the text 2.5 times larger than the base font of the system.

Ans. CSS :

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>.

CSS frameworks :

A CSS framework is a library allowing web design for easier, more standards-compliant using the Cascading Style Sheets language. Bootstrap, Foundation, Bulma, UIkit, Semantic UI are some of the CSS frameworks.

Different ways of using stylesheet :

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>
```

Program :

```
<HTML><HEAD><STYLE>
H1 {colour: red; font-family: arial; font-size: 2.50 em}
</STYLE></HEAD><BODY>
<H1> This is the H1 element </H1>
</BODY></HTML>
```

5. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. Discuss AJAX. Explain the application of AJAX with the help of suitable examples.

Ans.

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. 2.
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.

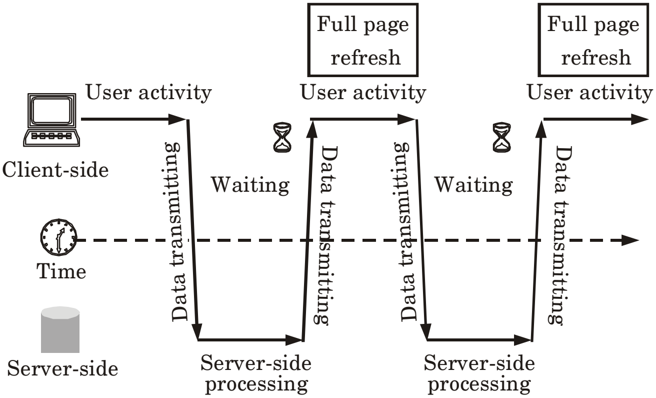


Fig. 2. Traditional web application communication flow.

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.

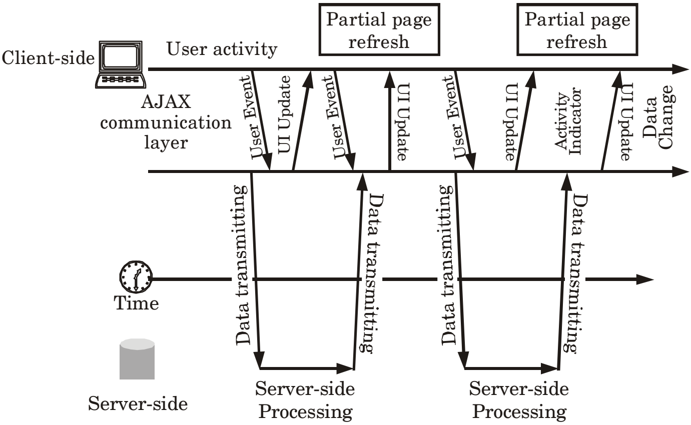


Fig. 3. AJAX style communication flow.

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.
4. Major application of AJAX is in login forms where user can enter their login details directly on the original page.

For example :

```
<!DOCTYPE html>
<html>
<style>
table,th,td {
border : 1px solid black;
border-collapse: collapse;
}
th,td {
padding: 5px;
}
</style>
<body>
<h1>The XMLHttpRequest Object</h1>
<form action=“”>
<select name=“customers”
onchange=“showCustomer(this.value)”>
<option value=“”>Select a customer:</option>
<option value=“name1”>Aakash Pandey</option>
<option value=“name2”>Mohan</option>
<option value=“name3”>Harshit Kumar</option>
</select>
</form>
<br>
<div id=“txtHint”>Customer info will be listed here...</div>
<script>
function showCustomer(str) {
 var xhttp;
if (str == “”) {
document.getElementById(“txtHint”).innerHTML = “”;
return;
}
xhttp = new XMLHttpRequest();
xhttp.onreadystatechange = function() {
if (this.readyState == 4 && this.status == 200) {
document.getElementById(“txtHint”).innerHTML=
this.responseText;
        }   };
xhttp.open(“GET”, “getcustomer.asp?q=”+str, true);
xhttp.send();
}
</script>
</body>
</html>
```

b. Compare Java and JavaScript. Explain and demonstrate 5 different types of objects in JavaScript with example.

Ans. **Comparison :**

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.

Different object 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)
{
      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.

For example :

```
<html>
<head>
<title>JavaScript Object </title>
</head>
<body>
<script type="text/javascript">
var value = Math.acos(-1);
document.write("ACOS Test Value : " + value +"<br>");
// Math object
var d = new Date();
document.write("<b>Locale String:</b> " +
d.toLocaleString()+"<br>");
// Date Object
var str = "CareerRide Info"; //String Object
var s = str.split();
document.write("<b>Char At:</b> " + str.charAt(1)+"<br>");
document.write("<b>CharCode At:</b> " +
str.charCodeAt(2)+"<br>");
document.write("<b>Index of:</b> " +
str.indexOf("ide")+"<br>");
</script>
</body>
</html>
```

Output:

```
ACOS Test Value : 3.141592653589793
Locale String: 22/02/2020 11:14:06 AM
Char At: a
CharCode At: 114
Index of: 7
```

6. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. Explain JDBC application architecture. List the various types of JDBC drivers. Discuss the steps to connect database with the web application using JDBC. Write a program to demonstrate how to retrieve the data from a table using JDBC API.

Ans. JDBC application architecture :

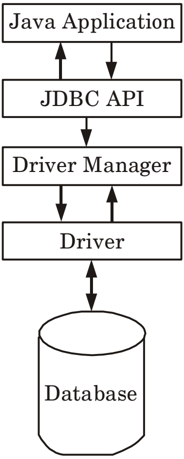


Fig. 4.

1. **Java application :** It is a standalone Java program which uses the JDBC API to get connected and perform operations on the database data.
2. **JDBC API :** It is a set of classes and interfaces used in a Java program for database operations. Java.sql and Javax.sql packages provide the necessary library support.
3. **Driver manager :** Java program uses DriverManager class to get the connection with the database.
4. **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.
5. **Database :** It is a collection of all enterprise data.

Types of JDBC driver :

1. **JDBC-ODBC bridge driver (Type 1 driver) :**
   a. These drivers are the bridge drivers such as JDBC-ODBC bridge.
   b. These drivers rely on an intermediary such as ODBC to transfer the SQL calls to the database.

c. Bridge drivers often rely on native code, although the JDBC-ODBC library native code is part of the Java-2 virtual machine.

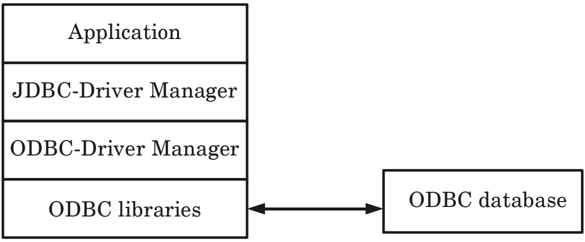


Fig. 5. JDBC-ODBC bridge driver.

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

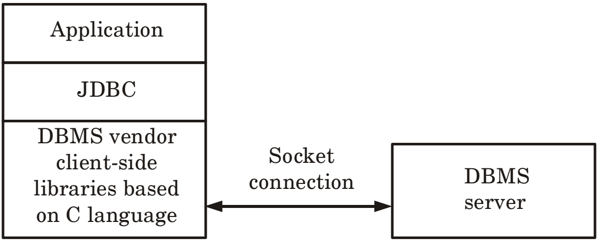


Fig. 6. Native API partly Java driver.

3. **JDBC net pure Java driver (Type 3 driver) :**
   a. JDBC net pure Java driver consists of JDBC and DBMS independent protocol driver.
   b. Here the calls are translated and sent to middle tier server through the socket.
   c. The middle tier contacts the database.
   d. Type 3 drivers call the database API on the server.
4. **Native protocol pure Java driver (Type 4 driver) :**
   a. A native protocol Java driver contains JDBC calls that are converted directly to the network protocol used by the DBMS server.
   b. This driver interacts directly with database.

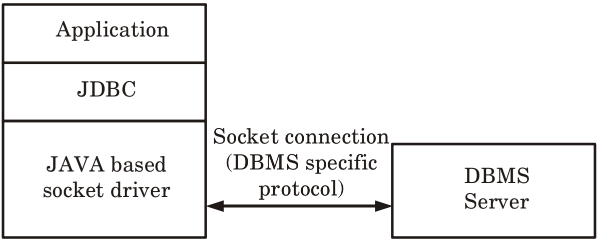


Fig. 7. Native protocol pure Java driver.

c. It does not require any native database library. So, it is also called thin driver.

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”);

Program :

```
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
```

b. Explain JavaBeans. Why they are used? Discuss setter and getter method with Java code.

Ans. Java Beans :

1. Java Beans are classes which encapsulate several objects into a single object.
2. It helps in accessing the objects from multiple places.
3. It is a portable, platform independent model written in Java.

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 constructor with no argument.
4. It can be manipulated visually in a builder tools.

Getter and setter method :

1. In Java, getter and setter are two conventional methods that are used for retrieving and updating value of a variable.

   For example :

   The following code is an example of simple class with a private variable and a couple of getter/setter methods :

```
public class SimpleGetterAndSetter {
private int number;
public int getNumber() {
return this.number;
}
public void setNumber(int num) {
this.number = num;
}
}
```

2. The class declares a private variable, number. Since "number" is private, code from outside this class cannot access the variable directly,

```
SimpleGetterAndSetter obj = new SimpleGetterAndSetter();
obj.number = 10;
int num = obj.number;
```

3. Inside the main class invoke the getter *i.e.*, getNumber() and the setter *i.e.*, setNumber() in order to read or update the variable,

For example :

```
public static void main (String args[ ])
{
SimpleGetterAndSetter obj = new SimpleGetterAndSetter();
obj.setNumber(10);
System.out.println (obj.getNumber ( ));
}
```

4. Getter and setter are also known as accessor and mutator in Java.

7. Attempt any **one** part of the following : **(7 × 1 = 7)**

a. Explain servlets with its life cycle. How its life cycle is different from the life cycle of JSP ? Explain with an example.

Ans. **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.

Life cycle of Servlet :

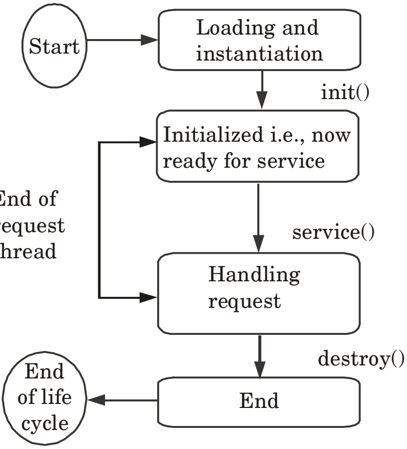


Fig. 8.

Stages of the Servlet life cycle :

1. **Loading a Servlet :**
   a. The first stage of the Servlet lifecycle involves loading and initializing the Servlet by the Servlet container.
   b. The Servlet container performs two operations in this stage :
      i. Loading : Loads the Servlet class.

ii. **Instantiation :** Creates an instance of the Servlet. To create a new instance of the Servlet, the container uses the no-argument constructor.

2. **Initializing a Servlet :**
   a. After the Servlet is instantiated successfully, the Servlet container initializes the instantiated Servlet object.
   b. The container initializes the Servlet object by invoking the Servlet.init(ServletConfig) method which accepts ServletConfig object reference as parameter.
3. **Handling request :**
   a. After initialization, the Servlet instance is ready to serve the client requests.
   b. The Servlet container performs the following operations when the Servlet instance is located to service a request :
      i. It creates the ServletRequest and ServletResponse objects.
      ii. After creating the request and response objects it invokes the Servlet.service(ServletRequest, ServletResponse) method by passing the request and response objects.
4. **Destroying a Servlet :**
   a. When a Servlet container decides to destroy the Servlet, it performs the following operations,
      i. It allows all the threads currently running in the service method of the Servlet instance to complete their jobs and get released.
      ii. After currently running threads have completed their jobs, the Servlet container calls the destroy() method on the Servlet instance.
   b. After the destroy() method is executed, the Servlet container releases all the references of this Servlet instance so that it becomes eligible for garbage collection.

Difference :

1. In servlet life cycle, the servlet object is created first.
2. The init() method is invoked by the servlet container and the servlet is initialized by its arguments.
3. Servlet's service() method is invoked next. At the end, the destroy() method is invoked.
4. In case of a Java Server Page life cycle, the .jsp is converted into .class file which is a servlet and then follows the process of the servlet. In other words, the .jsp is translated into servlet and the functionality is same as that of the servlet.

JSP lifecycle example :

Demo.jsp :

```
<html>
<head>
<title>Demo JSP</title>
```

```
</head>
<%
int demvar=0;%>
<body>
Count is :
<% Out.println(demovar++); %>
<body>
</html>
```

Demo JSP Page is converted into demo_jsp.servlet in the below code.

```
Public class demp_jsp extends HttpServlet{
Public void _jspservice(HttpServletRequest request, HttpServletResponse response) Throws IOException, ServletException
{
PrintWriter out = response.getWriter();
response.setContentType("text/html");
out.write ("<html><body>");
int demovar=0;
out.write("Count is:");
out.print(demovar++);
out.write("</body></html>");
}
}
```

5. The web container calls the init() method only once after creating the servlet instance.
6. The init() method is used to initialize the servlet. It is the life cycle method of the javax.servlet.Servlet interface.
7. Syntax of the init() method :

   ```
   public void init(ServletConfig config) throws ServletException
   ```
8. A new thread is then gets created, which invokes the_jspService() method, with a request (HttpServletRequest) and response (HttpServletResponse) objects as parameters shown below :

   ```
   [code language="java"]
   void _jspService( HttpServletRequest req, HttpServletResponse res)
   {
   //code goes here
   }[/code]
   ```
9. Invokes the jspDestroy() method to destroy the instance of the servlet class. code :

   ```
   [code language="java"]
   public void jspDestory()
   {
   //code to remove the instances of servlet class
   }
   ```

b. Discuss JSP in details. What are JSP directives ? Explain various types of directives with suitable example.

Ans. JSP :

1. 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.
2. JSP supports both scripting and element based dynamic content, and allows developers to create their own tag libraries.
3. JSP pages are compiled for efficient server processing.
4. JSP is platform independent. So, it can be easily upgraded or switched without affecting JSP based applications.

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:
<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/>
...
```

☺☺☺