**About Java**:

Initially the language was called as “oak” but it was renamed as “java” in 1995.The primary motivation of this language was the need for a platform-independent(i.e. architecture neutral)language that could be used to create software to be embedded in various consumer electronic devices.

* Java is a programmer’s language
* Java is cohesive and consistent
* Except for those constraint imposed by the Internet environment. Java gives the programmer, full control

Finally Java is to Internet Programming where c was to System Programming.

**Importance of Java to the Internet**

Java has had a profound effect on the Internet. This is because; java expands the Universe of objects that can move about freely in Cyberspace. In a network, two categories of objects are transmitted between the server and the personal computer. They are passive information and Dynamic active programs. in the areas of Security and probability. But Java addresses these concerns and by doing so, has opened the door to an exciting new form of program called the Applet.

**Applications and applets***.*

An application is a program that runs on our Computer under the operating system of that computer. It is more or less like one creating using C or C++ .Java’s ability to create Applets makes it important. An Applet I san application, designed to be transmitted over the Internet and executed by a Java-compatible web browser. An applet I actually a tiny Java program, dynamically downloaded across the network, just like an image. But the difference is, it is an intelligent program, not just a media file. It can be react to the user input and dynamically change.

**Java Architecture**

Java architecture provides a portable, robust, high performing environment for development. Java provides portability by compiling the byte codes for the Java Virtual Machine, which is then interpreted on each platform by the run-time environment. Java is a dynamic system, able to load code when needed from a machine in the same room or across the planet.

# Compilation of code

## When you compile the code, the Java compiler creates machine code (called byte code)for a hypothetical machine called Java Virtual Machine(JVM). The JVM is supposed t executed the byte code. The JVM is created for the overcoming the issue of probability. The code is written and compiled for one machine and interpreted on all machines .This machine is called Java Virtual Machine.

**Compiling and interpreting java source code.**

**Source code**

**Pc compiler**

**Macintosh compiler**

**SPARC Compiler**

**Java Byte code**

**Platform independent**

**Java interpreter**

**Java interpretermacintosh**

**)))**

**Java interpreter(SPARC)**

During run-time the Java interpreter tricks the byte code file into thinking that it is running on a Java Virtual Machine. In reality this could be an Intel Pentium windows 95 or sun SPARCstation running Solaris or Apple Macintosh running system and all could receive code from any computer through internet and run the Applets.

**Simple**:

Java was designed to be easy for the Professional programmer to learn and to use effectively. If you are an experienced C++ Programmer. Learning Java will oriented features of C++ . Most of the confusing concepts from C++ are either left out of Java or implemented in a cleaner, more approachable manner. In Java there are a small number of clearly defined ways to accomplish a given task.

### Object oriented

Java was not designed to be source-code compatible with any other language. This allowed the Java team the freedom to design with a blank state. One outcome of this was a clean usable, pragmatic approach to objects. The object model in Java is simple and easy to extend, while simple types, such as integers, are kept as high-performance non-objects.

### Robust

The multi-platform environment of the web places extraordinary demands on a program, because the program must execute reliably in a variety of systems. The ability to create robust programs. Was given a high priority in the design of Java. Java is strictly typed language; it checks your code at compile time and runtime.

Java virtually eliminates the problems of memory management and deal location, which is completely automatic. In a well-written Java program, all run-time errors can and should be managed by your program.

* + 1. **Servlets/JSP**

INTRODUCTION

A Servlet Is a generic server extension. a Java class that can be loaded

Dynamically to expand the functionality of a server.Servlets are commonly used with web servers. Where they can take the place CGI scripts.

A servlet is similar to proprietary server extension, except that it runs inside a Java Virtual Machine (JVM) on the server, so it is safe and portable

Servlets operate solely within the domain of the server.

Unlike CGI and Fast CGI, which use multiple processes to handle separate program or separate requests, separate threads within web server process handle all servlets. This means that servlets are all efficient and scalable.

Servlets are portable; both across operating systems and also across web servers. Java Servlets offer the best possible platform for web application development.

Servlets are used as replacement for CGI scripts on a web server, they can extend any sort of server such as a mail server that allows servelts t extend its functionality perhaps by performing a virus scan on all attached documents or handling mail filtering tasks.

Servlets provide a Java-based solution used to address the problems currently associated with doing server-side programming including inextensible scripting solutions platform-specific API’s and incomplete interface.

Servlets are objects that conform to a specific interface that can be plugged into a Java-based server.Servlets are to the server-side what applets are to the server-side what applets are to the client-side-object byte codes that can be dynamically loaded off the net. They differ form applets in than they are faceless objects(with out graphics or a GUI component).They serve as platform independent, dynamically loadable,plugable helper byte code objects on the server side that can be used to dynamically extend server-side functionality.

For example an HTTP servlet can be used to generate dynamic HTML content when you use servlets to do dynamic content you get the following advantages:

* They’re faster and cleaner then CGI scripts
* They use a standard API( the servlet API)
* They provide all the advantages of Java (run on a variety of servers without needing to be rewritten)

**A t t r a c t i v e n e s s o f S e r v l e t s:**

They are many features of servlets that make them easy and attractive to tuse these include:

* Easily configure using the GUI-based Admin tool]
* Can be Loaded and Invoked from a local disk or remotely across the network.
* Can be linked together or chained, so that on servlet can call another servlet, or several servlets in sequence.
* Can be called dynamically from with in HTML, pages using server-side include-tags.
* Are secure-even when downloading across the network, the servlet security model and servlet and box protect your system from unfriendly behavior.,

**Advantages of the servlet API**

One of the great advantages of the servlet API is protocol independent. It assumes nothing about:

* The protocol being used to transmit on the net
* How it is loaded
* The server environment it will be running in
* These quantities are important, because it allows the Servlet API to be embedded in many different kinds of servers. There are other advantages to the servelt API as well These include:
* It’s extensible-you can inherit all your functionality from the base classes made available to you
* It’s simple small, and easy to use.

**Features of Servlets:**

* Servlets are persistent.Servlet are loaded only by the web server and can maintain services between requests.
* Servlets are fast. Since servlets only need to be l\loaded once, they offer much better performance over their CGI counterparts.
* Servlets are platform independent.
* Servlets are extensible Java is a robust, object-oriented programming language, which easily can be extended to suit your needs.
* Servlets are secure
* Servlets are used with a variety of client.

Servlets are classes and interfaces from tow packages,javax .servlet and javax.servlet.http.The java.servlet package contains classes t support generic, protocol-independent servlets.The classes in the javax.servelt.http package To and HTTP specific functionality extend these classes

Every servlet must implement the javax.servelt interface.Most servlets implement it by extending one of two classes.javax.servlet.GenericServlet or javax.servlet.http.HttpServlet.A protocol-independent servlet should subclass Generic-Servlet.while an Http servlet should subclass HttpServlet, which is itself a subclass of Generic-servlet with added HTTP-specific functionality.

Unlike a java program, a servlet does not have a main() method,Instead the server in the process of handling requests invoke certain methods of a servlet.Each time the server dispatches a request to a servlet, it invokes the servelts Service() method,

A generic servlet should override its service() method to handle requests as appropriate for the servlet.The service() accepts two parameters a request object and a response object .The request object tells the servlet about the request, while the response object is used to return a response

InContrast.anHttp servlet usually does not override the service() method.Instead it overrides doGet() to handle GET requests and doPost() to handle Post requests. An Http servlet can override either or both of these modules the service() method of HttpServlet handles the setup and dispatching to all the doXXX() methods.which iswhy it usually should not be overridden

The remainders in the javax.servlet and javax.servlet.http.package are largely support classes .The ServletRequest and ServletResponse classes in javax.servlet provide access to generic server requests and responses while HttpServletRequest and HttpServletResponse classes in javax.servlet provide access to generic server requests and responses while HttpServletRequest and HttpServletResponse in javax.servlet.http provide access a HTTP requests and responses . The javax.servlet.http provide contains an HttpSession class that provides built-in session tracking functionality and Cookie class that allows quickly setup and processing HttpCookies.

**Loading Servlets:**

Servlets can be loaded from their places. From a directory that is on the CLASSPATH. The CLASSPATH of the JavaWebServer includes service root/classes/, which is where the system classes reside

From the <SERVICE\_ROOT/servlets/directory.This is not in the server’s classpath. A class loader is used to create servlets form this directory.New servlets can be added-existing servlets can be recompiled and the server will notice these changes. From a remote location.For this a code base like <http://nine.eng/classes/foo/> is required in addtion to the servlet’s class name.Refer to the admin Gui docs on servlet section to see how to set this up.

Loading Remote Servlets

Remote servlets can be loaded by:

* Configuring the admin Tool to setup automatic loading of remote servlets.
* Selectiong up server side include tags in .html files
* Defining a filter chain Configuration

**Invoking Servlets**

A servlet invoker is a servlet that invokes the “server” method on a named servlet.If the servlet is not loaded in the server,then the invoker first loades the servlet(either form local disk or from the network) and the then invokes the “service” method.Also like applets,local servlets in the server can be identified by just the class name.In other words, if a servlet name is not absolute.it is treated as local.

A Client can Invoke Servlets in the Following Ways:

* The client can ask for a document that is served by the servlet.
* The client(browser) can invoke the servlet directly using a URL, once it has been mapped using the SERVLET ALIASES Section of the admin GUI
* The servlet can be invoked through server side include tags.
* The servlet can be invoked by placing it in the servlets/directory
* The servlet can be invoked by using it in a filter chain

**The Servlet Life Cycle:-**

The Servlet life cycle is one of the most exciting features of Servlets.This life cycle is a powerful hybrid of the life cycles used in CGI programming and lower-level NSAPI and ISAPI programming.

The servlet life cycle allows servlet engines to address both the performance and resource problems of CGI and the security concents of low level server API programming.

Servlet life cycle is highly flexible Servers hava significant leeway in how they choose to support servlets.The only hard and fast rule is that a servlet engine must confor to the following life cycle contact:

* Create and initialize the servlets
* Handle zero or more service from clients
* Destroy the servlet and then garbage Collects it.

It’s perfectly legal for a servlet t be loaded, created an initialzed in its own JVM,only to be destroyed an dgarbage collected without hancdling any clientrequest or after handling just one request

The most common and most sensible life cycle implemntations for HTTP servelts are:

Single java virtual machine and astatine persistence.

**Init and Destroy**:-

Just like Applets servlets can define init() and destroy() methods, A servlets init(ServiceConfig) method is called by the server immediately after the server constructs the servlet’s instance.Depanding on the server and its configuration, this can be at any of these times

* When the server states
* When the servlet is first requested, just before the service() method is invoked
* At the request of the server administrator

In any case, nit() is guaranteed to be called before the servlet handles its first request

The init() method is typically used to perform servlet initialization creating or loading objects that are used by the servlet in handling of its request. In order to providing a new servlet any information about itself and its environment, a server has to call a servelts init() method and pass an object that implement the ServletConfig interface.

This ServletConfig object supplies a servlet with information about its initialization parameters.These parameters are given to the servlets and are not associated with any single request.They can specify initial values, such as where a counter should begin counting, or default values, perhaps a template to use when not specified by the request,

The server calls a servlet’s destroy() method when the servlet is about to be unloaded. In the destroy() method, a servlet should free any resources it has acquired that will not be garbage collected. The destroy() method also gives a servlet a chance to write out its unsaved. cached information or any persistent information that should be read during the next call to init().

**Session Tracking:**

HTTP is a stateless protocol, it provides no way for a server to recognize that a sequence of requests is all from the same client. This causes a problem for application such as shopping cart applications. Even in chat application server can’t know exactly who’s making a request of several clients.

The solution for this is for client to introduce itself as it makes each request, Each clients needs to provide a unique identifier that lets the server identify it, or it needs to give some information that the server can use to properly handle the request, There are several ways to send this introductory information with each request Such as:

**USER AUTHORIZATION:**

**One way to perform session tracking is to leverage the information that comes with**

User authorization. When a web server restricts access to some of its resources to only those clients that log in using a recognized username and password. After the client logs in, the username is available to a servlet through getRemoteUser ()

Wean use the username to track the session. Once a user has logged in, the browser remembers her username and resends the name and password as the user views new pages on the site. A servlet can identify the user through her username and they’re by

Track her session.

The biggest advantage of using user authorization to perform session tracking is that it’s easy to implement. Simply tell the protect a set of pages, and use getRemoteUser() to identify each client.Another advantage is that the technique works even when the user accesses your site form or exists her browser before coming back.

The biggest disadvantage of user authrization is that it requires each user to register for an account and then log in in each time the starts visiting your site. Most users will tolerate registering and lagging in as a necessary evil when they are accessing sensitive information, but its all overkill for simple session tracking.Other problem with user authorization is that a user cannot simultaneously maintain more than one session at the same site.

**Hidden Form Fields:**

One way to support anonymous session tracking is to use hidden from fields. As the name implies, these are fields added to an HTML, form that are not displayed in the client’s browser, They are sent back to the server when the form that contains them is submitted.

In a sense, hidden form fields define constant variables for a form. To a servlet receiving a submitted form, there is no difference between a hidden fields and a visible filed.

As more and more information is associated with a clients session . It can become burdensome to pass it all using hidden form fields. In these situations it’s possible to pass on just a unique session ID that identifies as particular clients session.

That session ID can be associated with complete information about its session that is stored on the server.

The advantage of hidden form fields is their ubiquity and support for anonymity. Hidden fields are supported in all the popular browsers, they demand on special server requirements, and they can be used with clients that haven’t registered or logged in.

The major disadvantage with this technique, however is that works only for a sequence of dynamically generated forms, The technique breaks down immediately with static documents, emailed documents book marked documents and browser shutdowns.

**URL Rewriting:**

URL rewriting is another way to support anonymous session tracking, With URL rewriting every local URL the user might click on is dynamically modified. or rewritten, to include extra, information. The extra information can be in the deform of extra path information, added parameters, or some custom, server-specific.URL change. Due to the limited space available in rewriting a URL, the extra information is usually limited to a unique session.

Each rewriting technique has its own advantage and disadvantage

Using extra path information works on all servers, and it works as a target for forms that use both the Get and Post methods. It does not work well if the servlet has to use the extra path information as true path information

The advantages and disadvantages of URL.rewriting closely match those of hidden form fileds,The major difference is that URL rewriting works for all dynamically created documents, such as the Help servlet, not just forms. With the right server support, custom URL rewriting can even work for static documents.

**Persistent Cookies:**

A fourth technique to perform session tracking involves persistent cookies. A cookie is a bit of information. sent by a web server to a browser that can later be read back form that browser. When a browser receives a cookie, it saves the cookie and there after sends the cookie back to the server each time it accesses a page on that server, subject to certain rules. Because a cookie’s value can uniquely identify a client, cookies are often used for session tracking.

Persistent cookies offer an elegant, efficient easy way to implement session tracking. Cookies provide as automatic an introduction for each request as we could hope for. For each request, a cookie can automatically provide a client’s session ID or perhaps a list of clients performance. The ability to customize cookies gives them extra power and versatility.

The biggest problem with cookies is that browsers don’t always accept cookies sometimes this is because the browser doesn’t support cookies. More often its because

The browser doesn’t support cookies. More often its because the user has specifically configured the browser to refuse cookies.

The power of serves:

The power of servlets is nothing but the advantages of servlets over other approaches, which include portability, power, efficiency, endurance, safety elegance, integration, extensibility and flexibility.

**Portability:**

As servlets are written in java and conform to a well defined and widely accepted API.they are highly portable across operating systems and across server implementation

We can develop a servlet on a windows NT machine running the java web server and later deploy it effortlessly on a high-end Unix server running apache. With servlets we can really “write once, serve every where”

Servlet portability is not the stumbling block it so often is with applets, for two reasons

First,Servlet portability is not mandatory i.e. servlets has to work only on server machines that we are using for development and deployment

Second, servlets avoid the most error-prone and inconstancy implemented portion of the java languages.

**Power:**

Servlets can harness the full power of the core java. API’s: such as Networking and Url access, multithreading, image manipulation, data compression, data base connectivity, internationalization, remote method invocation(RMI) CORBA connectivity, and object serialization, among others,

**Efficiency And Endurance:**

Servlet invocation is highly efficient, Once a servlet is loaded it generally remains in the server’s memory as a single object instance, There after the server invokes the servelt to handle a request using a simple, light weighted method invocation .Unlike the CGI, there’s no process to spawn or interpreter to invoke, so the servlet can begin handling the request almost immediately, Multiple, concurrent requests are handled the request almost immediately. Multiple, concurrent requests are handled by separate threads, so servlets are highly scalable.

Servlets in general are enduring objects. Because a servlets stays in the server’s memory as a single object instance. it automatically maintains its state and can hold onto external resources, such as database connections.

**Safety:**

Servlets support safe programming practices on a number of levels.

As they are written in java,servlets inherit the strong type safety of the java language. In addition the servlet API is implemented to be type safe. Java’s automatic garbage collection and lack of pointers mean that servlets are generally safe from memory management problems like dangling pointers invalid pointer references and memory leaks.

Servlets can handle errors safely, due to java’s exception – handling mechanism. If a servlet divides by zero or performs some illegal operations, it throws an exception that can be safely caught and handled by the server.

A server can further protect itself from servlets through the use of java security manager.A server can execute its servlets under the watch of a strict security manager.

**Elegance:**

The elegance of the servlet code is striking .Servlet code is clean, object oriented modular and amazingly simple one reason for this simplicity is the served API itself. Which includes methods and classes to handle many of the routine chores of servlet development. Even advanced to operations like cookie handling and session tracking tracking are abstracted int convenient classes.

**Integration:**

Servlets are tightly integrated with the server. This integration allows a servlet to cooperate with the server in two ways . for e.g.: a servlet can use the server to translate file paths, perform logging, check authorization, perform MIME type mapping and in some cases even add users to the server’s user database.

**Extensibility and Flexibility:**

The servlet API is designed to be easily extensible. As it stands today the API includes classes that are optimized for HTTP servlets.But later it can be extended and optimized for another type of servlets.It is also possible that its support for HTTP servlets could be further enhanced.

Servlets are also quite flexible, Sun also introduced java server pages. which offer a way to write snippets of servlet code directly with in a static HTML page using syntax similar to Microsoft’s Active server pages(ASP)

* + 1. **JDBC**

**What is JDBC?**

any relational database. One can write a single program using the JDBC API,and the JDBC is a Java Api for executing SQL,Statements(As a point of interest JDBC is trademarked name and is not an acronym; nevertheless,Jdbc is often thought of as standing for Java Database Connectivity. It consists of a set of classes and interfaces written in the Java Programming language.JDBC provides a standard API for tool/database developers and makes it possible to write database applications using a pure Java API

Using JDBC, it is easy to send SQL statements to virtually program will be able to send SQL .statements to the appropriate database. The Combination of Java and JDBC lets a programmer writes it once and run it anywhere.

**What Does JDBC Do?**

**Simply put,JDBC makes it possible to do three things**

* Establish a connection with a database
* Send SQL statements
* Process the results
* JDBC Driver Types
* The JDBC drivers that we are aware of this time fit into one of four categories
* JDBC-ODBC Bridge plus ODBC driver
* Native-API party-java driver
* JDBC-Net pure java driver
* Native-protocol pure Java driver

An individual database system is accessed via a specific JDBC driver that implements the java.sql.Driver interface. Drivers exist for nearly all-popular RDBMS systems, through few are available for free. Sun bundles a free JDBC-ODBC bridge driver with the JDK to allow access to a standard ODBC,data sources, such as a Microsoft Access database, Sun advises against using the bridge driver for anything other than development and very limited development.

JDBC drivers are available for most database platforms, from a number of vendors and in a number of different flavours. There are four driver categories

**Type 01-JDBC-ODBC Bridge Driver**

Type 01 drivers use a bridge technology to connect a java client to an ODBC database service. Sun’s JDBC-ODBC bridge is the most common type 01 driver. These drivers implemented using native code.

**Type 02-Native-API party-java Driver**

Type 02 drivers wrap a thin layer of java around database-specific native code libraries for Oracle databases, the native code libraries might be based on the OCI(Oracle call Interface) libraries, which were originally designed for **c/c++** programmers, Because type-02 drivers are implemented using native code. in some cases they have better performance than their all-java counter parts. They add an element of risk, however, because a defect in a driver’s native code section can crash the entire server

**Type 03-Net-Protocol All-Java Driver**

Type 03 drivers communicate via a generic network protocol to a piece of custom middleware. The middleware component might use any type of driver to provide the actual database access. These drivers are all java, which makes them useful for applet deployment and safe for servlet deployment

**Type-04-native-protocol All-java Driver**

Type o4 drivers are the most direct of the lot. Written entirely in java, Type 04 drivers understand database-specific networking. protocols and can access the database directly without any additional software

**JDBC-ODBC Bridge**

If possible use a Pure Java JDBC driver instead of the Bridge and an ODBC driver. This completely eliminates the client configuration required by ODBC.It also eliminates the potential that the Java VM could be corrupted by an error in the native code brought in by the Bridge(that is, the Bridge native library, the ODBC driver manager library, library, the ODBC driver library, and the database client library)

**WHAT IS The JDBC-ODBE Bridge ?**

The JDBC-ODBC Bridge is a Jdbc driver, which implements JDBC operations by translating them into ODBC operations. To ODBC it appears as a normal application program. The Bridge is implemented as the sun.jdbc.odbc Java package and contains a native library used to access ODBC.The Bridge is joint development of Intersolv and Java Soft

* + 1. **Oracle**

Oracle is a relational database management system, which organizes data in the form of tables. Oracle is one of many database servers based on RDBMS model, which manages a seer of data that attends three specific things-data structures, data integrity and data manipulation.

With oracle cooperative server technology we can realize the benefits of open, relational systems for all the applications. Oracle makes efficient use of all systems resources, on all hardware architecture; to deliver unmatched performance, price performance and scalability. Any DBMS to be called as RDBMS has to satisfy Dr.E.F.Codd’s rules.

**Features of Oracle:**

**Portable**

The Oracle RDBMS is available on wide range of platforms ranging from PCs to super computers and as a multi user loadable module for Novel NetWare, if you develop application on system you can run the same application on other systems without any modifications.

**Compatible**

Oracle commands can be used for communicating with IBM DB2 mainframe RDBMS that is different from Oracle, which is Oracle compatible with DB2. Oracle RDBMS is a high performance fault tolerant DBMS, which is specially designed for online transaction processing and for handling large database applications.

**Multithreaded Server Architecture**

Oracle adaptable multithreaded server architecture delivers scalable high performance for very large number of users on all hardware architecture including symmetric multiprocessors (sumps) and loosely coupled multiprocessors. Performance is achieved by eliminating CPU, I/O, memory and operating system bottlenecks and by optimizing the Oracle DBMS server code to eliminate all internal bottlenecks.

Oracle has become the most popular RDBMS in the market because of its ease of use

* Client/server architecture.
* Data independence.
* Ensuring data integrity and data security.
* Managing data concurrency.
* Parallel processing support for speed up data entry and online transaction processing used for applications.
* DB procedures, functions and packages.

**Dr.E.F.Codd’s Rules**

These rules are used for valuating a product to be called as relational database management systems. Out of 12 rules, a RDBMS product should satisfy at least 8 rules + rule called rule 0 that must be satisfied.

**RULE 0: Foundation Rule**

For any system to be advertised as, or claimed to be relational DBMS should manage database with in it self, with out using an external language.

**RULE 1: Information Rule**

All information in relational database is represented at logical level in only one way as values in tables.

**RULE 2: Guaranteed Access**

Each and every data in a relational database is guaranteed to be logically accessibility by using to a combination of table name, primary key value and column name.

**RULE 3: Systematic Treatment of Null Values**

Null values are supported for representing missing information and inapplicable information. They must be handled in systematic way, independent of data types.

**RULE 4: Dynamic Online Catalog based Relation Model**

The database description is represented at the logical level in the same way as ordinary data so that authorized users can apply the same relational language to its interrogation as they do to the regular data.

**RULE 5: Comprehensive Data Sub Language**

A relational system may support several languages and various models of terminal use. However there must be one language whose statement can express all of the following:

Data Definitions, View Definitions, Data Manipulations, Integrity, Constraints, Authorization and transaction boundaries.

**RULE 6: View Updating**

Any view that is theoretical can be updatable if changes can be made to the tables that effect the desired changes in the view.

**RULE 7: High level Update, Insert and Delete**

The capability of handling a base relational or derived relational as a single operand applies not only retrieval of data also to its insertion, updating, and deletion.

**RULE 8: Physical Data Independence**

Application program and terminal activities remain logically unimpaired whenever any changes are made in either storage representation or access method.

**RULE 9: Logical Data Independence**

Application programs and terminal activities remain logically unimpaired whenever any changes are made in either storage representation or access methods.

**RULE 10: Integrity Independence**

Integrity constraints specific to particular database must be definable in the relational data stored in the catalog, not in application program.

**RULE 11: Distributed Independence**

Whether or not a system supports database distribution, it must have a data sub-language that can support distributed databases without changing the application program.

**RULE 12: Non Sub-Version**

If a relational system has low level language, that low language cannot use to subversion or by pass the integrity rules and constraints expressed in the higher level relational language.

**Oracle supports the following Codd’s Rules**

Rule 1: Information Rule (Representation of information)-YES.

Rule 2: Guaranteed Access-YES.

Rule 3: Systematic treatment of Null values-YES.

Rule 4: Dynamic on-line catalog-based Relational Model-YES.

Rule 5: Comprehensive data sub language-YES.

Rule 6: View Updating-PARTIAL.

Rule 7: High-level Update, Insert and Delete-YES.

Rule 8: Physical data Independence-PARTIAL.

Rule 9: Logical data Independence-PARTIAL.

Rule 10: Integrity Independence-PARTIAL.

Rule 11: Distributed Independence-YES.

Rule 12: Non-subversion-YES.

* + 1. **HTML**

Hypertext Markup Language (HTML), the languages of the world wide web(WWW), allows users to produces web pages that included text, graphics and pointer to other web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML(Standard Generalized Markup Language),but

Specialized to hypertext and adapted to the Web. The idea behind Hypertext one point to another point. We can navigate through the information based on out interest and preference. A markup language is simply a series of items enclosed within the elements should be displayed.

Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

Html can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop

HTML provides tags(special codes) to make the document look attractive.

HTML provides are not case-sensitive. Using graphics,fonts,different sizes, color, etc.. can enhance the presentation of the document. Anything

That is not a tag is part of the document it self.

**Basic Html Tags**:

<!-- --> Specific Comments.

<A>………</A> Creates Hypertext links.

<B>………</B> Creates hypertext links.

<Big>……..</Big> Formats text in large-font

<Body>…….</Body> contains all tags and text in the Html-document

<Center>……</Center> Creates Text

<DD>………..</DD> Definition of a term.

<TABLE>……</TABLE> creates table

<Td>………..</Td> indicates table data in a table.

<Tr>………..</Tr> designates a table row

<Th>……….</Th> creates a heading in a table.

##### A D V A N T A G E S:-

* + A HTML document is small and hence easy to send over the net.It is small because it does not include formatted information.
  + HTML is platform independent

HTML tags are not case-sensitive.

**4.4.6 JAVA SCRIPT**

The Java Script Language

JavaScript is a compact , object-based scripting language for developing client and server internet applications. Netscape Navigator 2.0 interprets JavaScript statements embedded directly in an HTML page. and Livewire enables you to create server-based applications similar to common gateway interface(cgi) programs.

In a client application for Navigator, JavaScript statements embedded in an HTML Page can recognize and respond to user events such as mouse clicks form

Input, and page navigation.

For example, you can write a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code . Without any network transmission, an Html page with embedded Java Script can interpret the entered text and alert the user with a message dialog if the input is invalid or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.