Frequently asked JAVA(Core) Questions in Iterview

1)What is OOPs?

Ans: Object oriented programming organizes a program around its data,i.e.,objects and a set of well defined interfaces to that data.An object-oriented program can be characterized as data controlling access to code.

2)what is the difference between Procedural and OOPs?
Ans: a) In procedural program, programming logic follows certain procedures and the instructions are executed one after another. In OOPs program, unit of program is object, which is nothing but combination of data and code.
b) In procedural program,data is exposed to the whole program whereas in OOPs program,it is accessible with in the object and which in turn assures the security of the code.

3)What are Encapsulation, Inheritance and Polymorphism?
Ans: Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from
outside interference and misuse.
Inheritance is the process by which one object acquires the properties of another object.
Polymorphism is the feature that allows one interface to be used for general class actions.

4)What is the difference between Assignment and Initialization?
Ans: Assignment can be done as many times as desired whereas initialization can be done only once.

5)What are Class, Constructor and Primitive data types?
Ans: Class is a template for multiple objects with similar features and it is a blue print for objects. It defines a type of object according to the data the object can hold and the operations the object can perform.

Constructor is a special kind of method that determines how an object is initialized when created.

Primitive data types are 8 types and they are:
byte, short, int, long
float, double
boolean
char

6)What is an Object and how do you allocate memory to it?

Ans: Object is an instance of a class and it is a software unit that combines a structured set of data with a set of operations for inspecting and manipulating that data. When an object is created using new operator, memory is allocated to it.

7)What is the difference between constructor and method?
Ans: Constructor will be automatically invoked when an object is created whereas method has to be called explicitly.

8)What are methods and how are they defined?
Ans: Methods are functions that operate on instances of classes in which they are defined. Objects can communicate with each other using methods and can call methods in other classes.

Method definition has four parts. They are name of the method, type of object or primitive type the method

returns, a list of parameters and the body of the method. A method's signature is a combination of the first

three parts mentioned above.

9)What is the use of bin and lib in JDK?
Ans: Bin contains all tools such as javac, appletviewer, awt tool, etc., whereas lib contains API and all packages.

10)What is casting?
Ans: Casting is used to convert the value of one type to another.

11)How many ways can an argument be passed to a subroutine and explain them?
Ans: An argument can be passed in two ways. They are passing by value and passing by reference.

Passing by value: This method copies the value of an argument into the formal parameter of the subroutine.

Passing by reference: In this method, a reference to an argument (not the value of the argument) is passed to the parameter.

12)What is the difference between an argument and a parameter?
Ans: While defining method, variables passed in the method are called parameters. While using those methods, values passed to those variables are called arguments.

13)What are different types of access modifiers?
Ans: public: Any thing declared as public can be accessed from anywhere.

private: Any thing declared as private can't be seen outside of its class.

protected: Any thing declared as protected can be accessed by classes in the same package and subclasses in the other packages.

default modifier : Can be accessed only to classes in the same package.

14)What is final, finalize() and finally?
Ans: final : final keyword can be used for class, method and variables.

A final class cannot be subclassed and it prevents other programmers from subclassing a secure class to invoke insecure methods.

A final method can' t be overridden

A final variable can't change from its initialized value.

finalize( ) : finalize( ) method is used just before an object is destroyed and can be called just prior to

garbage collection.

finally : finally, a key word used in exception handling, creates a block of code that will be executed after a

try/catch block has completed and before the code following the try/catch block. The finally block will execute whether or not an exception is thrown.

for example, if a method opens a file upon exit, then you will not want the code that closes the file to be bypassed by the exception-handling mechanism. This finally keyword is designed to address this contingency.

15)What is UNICODE?
Ans: Unicode is used for internal representation of characters and strings and it uses 16 bits to represent each other.

16)What is Garbage Collection and how to call it explicitly?
Ans: When an object is no longer referred to by any variable, java automatically reclaims memory used by that object. This is known as garbage collection.

System.gc() method may be used to call it explicitly.

17)What is finalize() method ?
Ans: finalize () method is used just before an object is destroyed and can be called just prior to garbage collection.

18)What are Transient and Volatile Modifiers?
Ans: Transient: The transient modifier applies to variables only and it is not stored as part of its object's Persistent state. Transient variables are not serialized.

Volatile: Volatile modifier applies to variables only and it tells the compiler that the variable modified by volatile can be changed unexpectedly by other parts of the program.

19)What is method overloading and method overriding?
Ans: Method overloading: When a method in a class having the same method name with different arguments is said to be method overloading.

Method overriding : When a method in a class having the same method name with same arguments is said to be method overriding.

20)What is difference between overloading and overriding?
Ans: a) In overloading, there is a relationship between methods available in the same class whereas in overriding, there is relationship between a superclass method and subclass method.

b) Overloading does not block inheritance from the superclass whereas overriding blocks inheritance from the superclass.

c) In overloading, separate methods share the same name whereas in overriding,subclass method replaces the superclass.

d) Overloading must have different method signatures whereas overriding must have same signature.

21) What is meant by Inheritance and what are its advantages?
Ans: Inheritance is the process of inheriting all the features from a class. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.

22)What is the difference between this() and super()?
Ans: this() can be used to invoke a constructor of the same class whereas super() can be used to invoke a super class constructor.

23)What is the difference between superclass and subclass?
Ans: A super class is a class that is inherited whereas sub class is a class that does the inheriting.

24) What modifiers may be used with top-level class?
Ans: public, abstract and final can be used for top-level class.

25)What are inner class and anonymous class?
Ans: Inner class : classes defined in other classes, including those defined in methods are called inner classes.

An inner class can have any accessibility including private.

Anonymous class : Anonymous class is a class defined inside a method without a name and is instantiated and declared in the same place and cannot have explicit constructors.

26)What is a package?
Ans: A package is a collection of classes and interfaces that provides a high-level layer of access protection and name space management.

27) What is a reflection package?
Ans: java.lang.reflect package has the ability to analyze itself in runtime.

28) What is interface and its use?
Ans:Interface is similar to a class which may contain method's signature
only but not bodies and it is a formal set of method and constant declarations that must be defined by the class that implements it.
Interfaces are useful for:
a)Declaring methods that one or more classes are expected to implement
b)Capturing similarities between unrelated classes without forcing a class relationship.

c)Determining an object's programming interface without revealing the actual body of the class.

29) What is an abstract class?
Ans: An abstract class is a class designed with implementation gaps for subclasses to fill in and is deliberately incomplete.

30) What is the difference between Integer and int?
Ans: a) Integer is a class defined in the java.lang package, whereas int is a primitive data type defined in the Java language itself. Java does not automatically convert from one to the other.
b) Integer can be used as an argument for a method that requires an object, whereas int can be used for calculations.

31) What is a cloneable interface and how many methods does it contain?
Ans- It is not having any method because it is a TAGGED or MARKER interface.

32) What is the difference between abstract class and interface?
Ans: a) All the methods declared inside an interface are abstract whereas abstract class must have at least one abstract method and others may be concrete or abstract.
b) In abstract class, key word abstract must be used for the methods whereas interface we need not use that keyword for the methods.
c) Abstract class must have subclasses whereas interface can't have subclasses.

33) Can you have an inner class inside a method and what variables can you access?
Ans: Yes, we can have an inner class inside a method and final variables can be accessed.

34) What is the difference between String and String Buffer?
Ans: a) String objects are constants and immutable whereas StringBuffer objects are not.
b) String class supports constant strings whereas StringBuffer class supports growable and modifiable strings.

35) What is the difference between Array and vector?
Ans: Array is a set of related data type and static whereas vector is a growable array of objects and dynamic.

36) What is the difference between exception and error?
Ans: The exception class defines mild error conditions that your program encounters.
Ex: Arithmetic Exception, FilenotFound exception
Exceptions can occur when try to open the file, which does not exist the network connection is disrupted
-- operands being manipulated are out of prescribed ranges
-- the class file you are interested in loading is missing
The error class defines serious error conditions that you should not attempt to recover from. In most cases it is advisable to let the program terminate when such an error is encountered.

Ex: Running out of memory error, Stack overflow error.

37) What is the difference between process and thread?
Ans: Process is a program in execution whereas thread is a separate path of execution in a program.

38) What is multithreading and what are the methods for inter-thread communication and what is the class in which these methods are defined?
Ans: Multithreading is the mechanism in which more than one thread run independent of each other within the process.
wait (), notify () and notifyAll() methods can be used for inter-thread communication and these methods are in Object class.
wait( ) : When a thread executes a call to wait( ) method, it surrenders the object lock and enters into a waiting state.
notify( ) or notifyAll( ) : To remove a thread from the waiting state, some other thread must make a call to notify( ) or notifyAll( ) method on the same object.

39) What is the class and interface in java to create thread and which is the most advantageous method?
Ans: Thread class and Runnable interface can be used to create threads and using Runnable interface is the most advantageous method to create threads because we need not extend thread class here.

40) What are the states associated in the thread?
Ans: Thread contains ready, running, waiting and dead states.

41) What is synchronization?
Ans: Synchronization is the mechanism that ensures that only one thread is accessed the resources at a time.

42) When you will synchronize a piece of your code?
Ans: When you expect your code will be accessed by different threads and these threads may change a particular data causing data corruption.

43) What is deadlock?
Ans: When two threads are waiting each other and can't precede the program is said to be deadlock.

44) What is daemon thread and which method is used to create the daemon thread?
Ans: Daemon thread is a low priority thread which runs intermittently in the back ground doing the garbage collection operation for the java runtime system. setDaemon method is used to create a daemon thread.

45) Are there any global variables in Java, which can be accessed by other part of your program?

Ans: No, it is not the main method in which you define variables. Global variables is not possible because concept of encapsulation is eliminated here.

46)What is an applet?
Ans: Applet is a dynamic and interactive program that runs inside a web page displayed by a java capable browser.

47)What is the difference between applications and applets?
Ans: a)Application must be run on local machine whereas applet needs no explicit installation on local machine.
b)Application must be run explicitly within a java-compatible virtual machine whereas applet loads and runs itself automatically in a java-enabled browser.
d)Application starts execution with its main method whereas applet starts execution with its init method.
e)Application can run with or without graphical user interface whereas applet must run within a graphical user interface.

48)How does applet recognize the height and width?
Ans:Using getParameters() method.

49)When do you use codebase in applet?
Ans:When the applet class file is not in the same directory, codebase is used.

50)What is the lifecycle of an applet?
Ans:init( ) method - Can be called when an applet is first loaded
start( ) method - Can be called each time an applet is started
paint( ) method - Can be called when the applet is minimized or maximized
stop( ) method - Can be used when the browser moves off the applet's page
destroy( ) method - Can be called when the browser is finished with the applet

51)How do you set security in applets?
Ans: using setSecurityManager() method

52) What is an event and what are the models available for event handling?
Ans: An event is an event object that describes a state of change in a source. In other words, event occurs when an action is generated, like pressing button, clicking mouse, selecting a list, etc.

There are two types of models for handling events and they are:
a) event-inheritance model and b) event-delegation model

53) What are the advantages of the model over the event-inheritance model?
Ans: The event-delegation model has two advantages over the event-inheritance model. They are:
a)It enables event handling by objects other than the ones that generate the events. This allows a clean separation between a component's design and its use.

b)It performs much better in applications where many events are generated. This performance improvement is due to the fact that the event-delegation model does not have to be repeatedly process unhandled events as is the case of the event-inheritance.

54)What is source and listener ?
Ans: source : A source is an object that generates an event. This occurs when the internal state of that object changes in some way.

listener : A listener is an object that is notified when an event occurs. It has two major requirements. First, it must have been registered with one or more sources to receive notifications about specific types of events.Second, it must implement methods to receive and process these notifications.

55) What is adapter class?
Ans: An adapter class provides an empty implementation of all methods in an event listener interface. Adapter classes are useful when you want to receive and process only some of the events that are handled by a particular event listener interface. You can define a new class to act listener by extending one of the adapter classes and implementing only those events in which you are interested.

For example, the MouseMotionAdapter class has two methods, mouseDragged( )and mouseMoved(). The signatures of these empty are exactly as defined in the MouseMotionListener interface. If you are interested in only mouse drag events, then you could simply extend MouseMotionAdapter and implement mouseDragged( ) .

56)What is meant by controls and what are different types of controls in AWT?
Ans: Controls are components that allow a user to interact with your application and the AWT supports the following types of controls:

Labels, Push Buttons, Check Boxes, Choice Lists, Lists, Scrollbars, Text Components.These controls are subclasses of Component.

57) What is the difference between choice and list?
Ans: A Choice is displayed in a compact form that requires you to pull it down to see the list of available choices and only one item may be selected from a choice.A List may be displayed in such a way that several list items are visible and it supports the selection of one or more list items.

58) What is the difference between scrollbar and scrollpane?
Ans: A Scrollbar is a Component, but not a Container whereas Scrollpane is a Conatiner and handles its own events and perform its own scrolling.

59) What is a layout manager and what are different types of layout managers available in java.awt?
Ans: A layout manager is an object that is used to organize components in a container. The different layouts are available are FlowLayout, BorderLayout, CardLayout, GridLayout and GridBagLayout.

60) How are the elements of different layouts organized?
Ans: FlowLayout: The elements of a FlowLayout are organized in a top to bottom, left to right fashion.

BorderLayout: The elements of a BorderLayout are organized at the borders (North, South, East and West) and the center of a container.

CardLayout: The elements of a CardLayout are stacked, on top of the other, like a deck of cards.

GridLayout: The elements of a GridLayout are of equal size and are laid out using the square of a grid.

GridBagLayout: The elements of a GridBagLayout are organized according to a grid.

However, the elements are of different size and may occupy more than one row or column of the grid. In addition, the rows and columns may have different sizes.

61) Which containers use a Border layout as their default layout?
Ans: Window, Frame and Dialog classes use a BorderLayout as their layout.

62) Which containers use a Flow layout as their default layout?
Ans: Panel and Applet classes use the FlowLayout as their default layout.

63) What are wrapper classes?
Ans: Wrapper classes are classes that allow primitive types to be accessed as objects.

64) What are Vector, Hashtable, LinkedList and Enumeration?
Ans: Vector : The Vector class provides the capability to implement a growable array of objects.

Hashtable : The Hashtable class implements a Hashtable data structure. A Hashtable indexes and stores objects in a dictionary using hash codes as the object's keys. Hash codes are integer values that identify objects.

LinkedList: Removing or inserting elements in the middle of an array can be done using LinkedList. A LinkedList stores each object in a separate link whereas an array stores object references in consecutive locations.

Enumeration: An object that implements the Enumeration interface generates a series of elements, one at a time. It has two methods, namely hasMoreElements( ) and nextElement( ). HasMoreElemnts( ) tests if this enumeration has more elements and nextElement method returns successive elements of the series.

65) What is the difference between set and list?
Ans: Set stores elements in an unordered way but does not contain duplicate elements, whereas list stores elements in an ordered way but may contain duplicate elements.

66) What is a stream and what are the types of Streams and classes of the Streams?
Ans: A Stream is an abstraction that either produces or consumes information. There are two types of Streams and they are:

Byte Streams: Provide a convenient means for handling input and output of bytes.

Character Streams: Provide a convenient means for handling input & output of characters.

Byte Streams classes: Are defined by using two abstract classes, namely InputStream and OutputStream.

Character Streams classes: Are defined by using two abstract classes, namely Reader and Writer.

67) What is the difference between Reader/Writer and InputStream/Output Stream?
Ans: The Reader/Writer class is character-oriented and the InputStream/OutputStream class is byte-oriented.

68) What is an I/O filter?
Ans: An I/O filter is an object that reads from one stream and writes to another, usually altering the data in some way as it is passed from one stream to another.

69) What is serialization and deserialization?
Ans: Serialization is the process of writing the state of an object to a byte stream.
Deserialization is the process of restoring these objects.

70) What is JDBC?
Ans: JDBC is a set of Java API for executing SQL statements. This API consists of a set of classes and interfaces to enable programs to write pure Java Database applications.

71) What are drivers available?
Ans: a)JDBC-ODBC Bridge driver
b)Native API Partly-Java driver
c)JDBC-Net Pure Java driver
d)Native-Protocol Pure Java driver

72) What is the difference between JDBC and ODBC?
Ans: a) OBDC is for Microsoft and JDBC is for Java applications.
b) ODBC can't be directly used with Java because it uses a C interface.
c) ODBC makes use of pointers which have been removed totally from Java.
d) ODBC mixes simple and advanced features together and has complex options for simple queries. But JDBC is designed to keep things simple while allowing advanced capabilities when required.
e) ODBC requires manual installation of the ODBC driver manager and driver on all client machines. JDBC drivers are written in Java and JDBC code is automatically installable, secure, and portable on all platforms.
f) JDBC API is a natural Java interface and is built on ODBC. JDBC retains some of the basic features of ODBC.

73) What are the types of JDBC Driver Models and explain them?
Ans: There are two types of JDBC Driver Models and they are:
a)two tier model and b) Three tier model

Two tier model: In this model, Java applications interact directly with the database. A JDBC driver is required to communicate with the particular database management system that is being accessed. SQL statements are sent to the database and the results are given to user. This model is referred to as client/server configuration where user is the client and the machine that has the database is called as the server.
Three tier model: A middle tier is introduced in this model. The functions of this model are:
a)Collection of SQL statements from the client and handing it over to the database,
b)Receiving results from database to the client and
c)Maintaining control over accessing and updating of the above.

74) What are the steps involved for making a connection with a database or how do you connect to a database?
Ans: a)Loading the driver : To load the driver, Class.forName( ) method is used.

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

When the driver is loaded, it registers itself with The java.sql.DriverManager class as an available database driver.
b) Making a connection with database : To open a connection to a given database, DriverManager.getConnection( ) method is used.

```
Connection con = DriverManager.getConnection ("jdbc:odbc:somedb", "user",
"password");
```

c) Executing SQL statements : To execute a SQL query, java.sql.statements class is used.
createStatement( ) method of Connection to obtain a new Statement object.

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

A query that returns data can be executed using the executeQuery( ) method of Statement. This method executes the statement and returns a java.sql.ResultSet that encapsulates the retrieved data:

```
ResultSet rs = stmt.executeQuery("SELECT * FROM some table");
```

d) Process the results : ResultSet returns one row at a time. Next( ) method of ResultSet object can be called to move to the next row. The getString( ) and getObject( ) methods are used for retrieving column values:

```
while(rs.next( ) ) {
   String event = rs.getString("event");
   Object count = (Integer) rs.getObject("count");
```

75) What type of driver did you use in project?
Ans: JDBC-ODBC Bridge driver (is a driver that uses native(C language) libraries and makes calls to an existing ODBC driver to access a database engine).

76) What are the types of statements in JDBC?
Ans: Statement -- To be used createStatement() method for executing single SQL statement
PreparedStatement -- To be used preparedStatement() method for executing same SQL statement over and
over

CallableStatement -- To be used prepareCall( ) method for multiple SQL statements over and over

77) What is stored procedure?

Ans: Stored procedure is a group of SQL statements that forms a logical unit and performs a particular task.

Stored Procedures are used to encapsulate a set of operations or queries to execute on database. Stored procedures can be compiled and executed with different parameters and results and may have any combination of input/output parameters.

78) How to create and call stored procedures?

Ans: To create stored procedures:

```
Create procedure procedurename (specify in, out and in out parameters)
BEGIN
Any multiple SQL statement;
END;
```

To call stored procedures:

```
CallableStatement  csmt  = con.prepareCall("{call procedure name(?,?)}");
csmt.registerOutParameter(column no., data type);
csmt.setInt(column no., column name)
csmt.execute( );
```

79) What is servlet?

Ans: Servlets are modules that extend request/response-oriented servers, such as java-enabled web servers.

For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.

80) What are the classes and interfaces for servlets?

Ans: There are two packages in servlets and they are javax.servlet and javax.servlet.http.

Javax.servlet contains:

Interfaces	Classes
Servlet	Generic Servlet
ServletRequest	ServletInputStream
ServletResponse	ServletOutputStream
ServletConfig	ServletException
ServletContext	UnavailableException
SingleThreadModel	

Javax.servlet.http contains:

Interfaces	Classes
HttpServletRequest	Cookie
HttpServletResponse	HttpServlet
HttpSession	HttpSessionBindingEvent
HttpSessionCintext	HttpUtils
HttpSeesionBindingListener	

81) What is the difference between an applet and a servlet?

Ans: a) Servlets are to servers what applets are to browsers.

b) Applets must have graphical user interfaces whereas servlets have no graphical user interfaces.

82) What is the difference between doPost and doGet methods?

Ans: a) doGet() method is used to get information, while doPost( ) method is used for posting information.

b) doGet() requests can't send large amount of information and is limited to 240-255 characters. However,

doPost( )requests passes all of its data, of unlimited length.

c) A doGet( ) request is appended to the request URL in a query string and this allows the exchange is

visible to the client, whereas a doPost() request passes directly over the socket connection as part of its

HTTP request body and the exchange are invisible to the client.

83) What is the life cycle of a servlet?

Ans: Each Servlet has the same life cycle:

a) A server loads and initializes the servlet by init () method.

b) The servlet handles zero or more client's requests through service( ) method.

c) The server removes the servlet through destroy() method.

84) Who is loading the init() method of servlet?

Ans: Web server

85) What are the different servers available for developing and deploying Servlets?

Ans: a) Java Web Server

b) JRun

g) Apache Server

h) Netscape Information Server

i) Web Logic

86) How many ways can we track client and what are they?

Ans: The servlet API provides two ways to track client state and they are:

a) Using Session tracking and b) Using Cookies.

87) What is session tracking and how do you track a user session in servlets?

Ans: Session tracking is a mechanism that servlets use to maintain state about a series requests
from the same user across some period of time. The methods used for session tracking are:

a) User Authentication - occurs when a web server restricts access to some of its resources to only those clients that log in using a recognized username and password

b) Hidden form fields - fields are added to an HTML form that are not displayed in the client's browser. When the form containing the fields is submitted, the fields are sent back to the server

c) URL rewriting - every URL that the user clicks on is dynamically modified or rewritten to include extra information. The extra information can be in the form of extra path information, added parameters or some custom, server-specific URL change.

d) Cookies - a bit of information that is sent by a web server to a browser and which can later be read back from that browser.

e) HttpSession- places a limit on the number of sessions that can exist in memory. This limit is set in the session.maxresidents property

88) What is Server-Side Includes (SSI)?

Ans: Server-Side Includes allows embedding servlets within HTML pages using a special servlet tag. In many
servlets that support servlets, a page can be processed by the server to include output from servlets at certain
points inside the HTML page. This is accomplished using a special internal SSINCLUDE, which processes
the servlet tags. SSINCLUDE servlet will be invoked whenever a file with an. shtml extension is requested.
So HTML files that include server-side includes must be stored with an .shtml extension.

89) What are cookies and how will you use them?

Ans: Cookies are a mechanism that a servlet uses to have a client hold a small amount of state-information
associated with the user.

a) Create a cookie with the Cookie constructor:
public Cookie(String name, String value)

b) A servlet can send a cookie to the client by passing a Cookie object to the addCookie() method of
HttpServletResponse:
public void HttpServletResponse.addCookie(Cookie cookie)

c) A servlet retrieves cookies by calling the getCookies() method of HttpServletRequest:
public Cookie[ ] HttpServletRequest.getCookie( ).

90) Is it possible to communicate from an applet to servlet and how many ways and how?
Ans: Yes, there are three ways to communicate from an applet to servlet and they are:

a) HTTP Communication(Text-based and object-based)
b) Socket Communication
c) RMI Communication

(You can say, by using URL object open the connection to server and get the InputStream from URLConnection object).

Steps involved for applet-servlet communication:

1) Get the server URL.
URL url = new URL();

2) Connect to the host
URLConnection Con = url.openConnection();

3) Initialize the connection
Con.setUseCatches(false):
Con.setDoOutput(true);
Con.setDoInput(true);

4) Data will be written to a byte array buffer so that we can tell the server the length of the data.
ByteArrayOutputStream byteout = new ByteArrayOutputStream();

5) Create the OutputStream to be used to write the data to the buffer.
DataOutputStream out = new DataOutputStream(byteout);

91) What is connection pooling?
Ans: With servlets, opening a database connection is a major bottleneck because we are creating and tearing down a new connection for every page request and the time taken to create connection will be more.

Creating a connection pool is an ideal approach for a complicated servlet. With a connection pool, we can duplicate only the resources we need to duplicate rather than the entire servlet. A connection pool can also intelligently manage the size of the pool and make sure each connection remains valid. A number of connection pool packages are currently available. Some like DbConnectionBroker are freely available from Java Exchange Works by creating an object that dispenses connections and connection Ids on request.

The ConnectionPool class maintains a Hastable, using Connection objects as keys and Boolean values as stored values. The Boolean value indicates whether a connection is in use or not. A program calls getConnection( ) method of the ConnectionPool for getting Connection object it can use; it calls returnConnection( ) to give the connection back to the pool.

92) Why should we go for interservlet communication?

Ans: Servlets running together in the same server communicate with each other in several ways.

The three major reasons to use interservlet communication are:

a) Direct servlet manipulation - allows to gain access to the other currently loaded servlets and perform certain tasks (through the ServletContext object)

b) Servlet reuse - allows the servlet to reuse the public methods of another servlet.

c) Servlet collaboration - requires to communicate with each other by sharing specific information (through method invocation)

93) Is it possible to call servlet with parameters in the URL?

Ans: Yes. You can call a servlet with parameters in the syntax as (?Param1 = xxx || m2 = yyy).

94) What is Servlet chaining?

Ans: Servlet chaining is a technique in which two or more servlets can cooperate in servicing a single request.

In servlet chaining, one servlet's output is piped to the next servlet's input. This process continues until the

last servlet is reached. Its output is then sent back to the client.

95) How do servlets handle multiple simultaneous requests?

Ans: The server has multiple threads that are available to handle requests. When a request comes in, it is

assigned to a thread, which calls a service method (for example: doGet(), doPost( ) and service( )) of the

servlet. For this reason, a single servlet object can have its service methods called by many threads at once.

96) What is the difference between TCP/IP and UDP?

Ans: TCP/IP is a two-way communication between the client and the server and it is a reliable and there is a confirmation regarding reaching the message to the destination. It is like a phone call.

UDP is a one-way communication only between the client and the server and it is not a reliable and there is no confirmation regarding reaching the message to the destination. It is like a postal mail.

97) What is Inet address?

Ans: Every computer connected to a network has an IP address. An IP address is a number that uniquely identifies each computer on the Net. An IP address is a 32-bit number.

98) What is Domain Naming Service(DNS)?

Ans: It is very difficult to remember a set of numbers(IP address) to connect to the Internet. The Domain Naming Service(DNS) is used to overcome this problem. It maps one particular IP address to a string of characters.

For example, www.mascom.com implies com is the domain name reserved for US commercial sites, moscom is the name of the company and www is the name of the specific computer, which is mascom's server.

99) What is URL?
Ans: URL stands for Uniform Resource Locator and it points to resource files on the Internet.

URL has four components:

http://www.Pentafour.com:80/index.html

http - protocol name, Pentafour - IP address or host name, 80 - port number and index.html - file path.

100) What is RMI and steps involved in developing an RMI object?
Ans: Remote Method Invocation (RMI) allows java object that executes on one machine and to invoke the method

of a Java object to execute on another machine.

The steps involved in developing an RMI object are:

a) Define the interfaces
b) Implementing these interfaces
c) Compile the interfaces and their implementations with the java compiler
d) Compile the server implementation with RMI compiler
e) Run the RMI registry
f) Run the application

101) What is RMI architecture?
Ans: - RMI architecture consists of four layers and each layer performs specific functions:

a) Application layer ---- contains the actual object definition
b) Proxy layer ---- consists of stub and skeleton
c) Remote Reference layer ---- gets the stream of bytes from the transport layer and sends it to the proxy
layer
d) Transportation layer ---- responsible for handling the actual machine-to-machine communication

102) what is UnicastRemoteObject?
Ans: All remote objects must extend UnicastRemoteObject, which provides functionality that is needed to make

objects available from remote machines.

103) Explain the methods, rebind( ) and lookup() in Naming class?
Ans: rebind( ) of the Naming class(found in java.rmi) is used to update the RMI registry on the server machine.

Naming. rebind("AddSever", AddServerImpl);

lookup( ) of the Naming class accepts one argument, the rmi URL and returns a reference to an object of type AddServerImpl.

104) What is a Java Bean?
Ans: A Java Bean is a software component that has been designed to be reusable in a variety of different environments.

105) What is a Jar file?
Ans: Jar file allows to efficiently deploying a set of classes and their associated resources. The elements in a jar file are compressed, which makes downloading a Jar file much faster than separately downloading several uncompressed files.

The package java.util.zip contains classes that read and write jar files.

106) What is BDK?
Ans: BDK, Bean Development Kit is a tool that enables to create, configure and connect a set of set of Beans and it can be used to test Beans without writing a code.

107) What is JSP?
Ans: JSP is a dynamic scripting capability for web pages that allows Java as well as a few special tags to be embedded into a web file (HTML/XML, etc). The suffix traditionally ends with .jsp to indicate to the web server that the file is a JSP files. JSP is a server side technology - you can't do any client side validation with it.

The advantages are:

a) The JSP assists in making the HTML more functional. Servlets on the other hand allow outputting of HTML but it is a tedious process.

b) It is easy to make a change and then let the JSP capability of the web server you are using deal with compiling it into a servlet and running it.

108) What are JSP scripting elements?
Ans: JSP scripting elements lets to insert Java code into the servlet that will be generated from the current JSP page. There are three forms:

a) Expressions of the form <%= expression %> that are evaluated and inserted into the output,

b) Scriptlets of the form <% code %> that are inserted into the servlet's service method, and

c) Declarations of the form <%! Code %> that are inserted into the body of the servlet class, outside of any existing methods.

109) What are JSP Directives?
Ans: A JSP directive affects the overall structure of the servlet class. It usually has the following form:

```
   <%@ directive attribute="value" %>
```

However, you can also combine multiple attribute settings for a single directive, as follows:

```
   <%@ directive  attribute1="value1"
             attribute 2="value2"
            ...
            attributeN ="valueN" %>
```

There are two main types of directive: page, which lets to do things like import classes, customize the servlet   superclass, and the like; and include, which lets to insert a file into the servlet class at the time the JSP file is translated into a servlet

110) What are Predefined variables or implicit objects?
Ans: To simplify code in JSP expressions and scriptlets, we can use eight automatically defined variables,
sometimes called implicit objects. They are request, response, out, session, application, config,
pageContext, and page.

111) What are JSP ACTIONS?
Ans: JSP actions use constructs in XML syntax to control the behavior of the servlet engine. You can dynamically insert a file, reuse JavaBeans components, forward the user to another page, or generate HTML for the Java plugin. Available actions include:

? jsp:include - Include a file at the time the page is requested.
? jsp:useBean - Find or instantiate a JavaBean.
? jsp:setProperty - Set the property of a JavaBean.
? jsp:getProperty - Insert the property of a JavaBean into the output.
? jsp:forward - Forward the requester to a newpage.
? Jsp: plugin - Generate browser-specific code that makes an OBJECT or EMBED

112) How do you pass data (including JavaBeans) to a JSP from a servlet?
Ans: (1) Request Lifetime: Using this technique to pass beans, a request dispatcher (using either "include" or
forward") can be called. This bean will disappear after processing this request has been completed.
Servlet:

```
   request.setAttribute("theBean", myBean);
   RequestDispatcher rd =   getServletContext().getRequestDispatcher("thepage.jsp");
   rd.forward(request, response);
```

JSP PAGE:

```
   <jsp: useBean id="theBean" scope="request" class="....." />
```

(2) Session Lifetime: Using this technique to pass beans that are relevant to a particular session (such as in individual user login) over a number of requests. This bean will disappear when the session is invalidated or it times out, or when you remove it.

Servlet:

```
   HttpSession session = request.getSession(true);
   session.putValue("theBean", myBean);
   /* You can do a request dispatcher here,
      or just let the bean be visible on the
      next request */
```

JSP Page:

```
   <jsp:useBean id="theBean" scope="session" class="..." />
```

3) Application Lifetime: Using this technique to pass beans that are relevant to all servlets and JSP pages in a particular app, for all users. For example, I use this to make a JDBC connection pool object available to the various servlets and JSP pages in my apps. This bean will disappear when the servlet engine is shut down, or when you remove it.

Servlet:

```
   GetServletContext(). setAttribute("theBean", myBean);
```

JSP PAGE:

```
   <jsp:useBean id="theBean" scope="application" class="..." />
```

113) How can I set a cookie in JSP?

Ans: response.setHeader("Set-Cookie", "cookie string");

To give the response-object to a bean, write a method setResponse (HttpServletResponse response)
- to the bean, and in jsp-file:

```
<%
bean.setResponse (response);
%>
```

114) How can I delete a cookie with JSP?

Ans: Say that I have a cookie called "foo," that I set a while ago & I want it to go away. I simply:

```
<%
 Cookie killCookie = new Cookie("foo", null);
  KillCookie.setPath("/");
  killCookie.setMaxAge(0);
  response.addCookie(killCookie);
%>
```

115) How are Servlets and JSP Pages related?

Ans: JSP pages are focused around HTML (or XML) with Java codes and JSP tags inside them. When a web server that has JSP support is asked for a JSP page, it checks to see if it has already compiled the page into a servlet. Thus, JSP pages become servlets and are transformed into pure Java and then compiled, loaded into the server and executed.