**Question 1**

Give names of the new features of JDK 1.5/1.6 as compared to JDK 1.4.

Generics

Enhanced for Loop

Autoboxing/Unboxing

Typesafe Enums

Varargs

Static Import

Metadata (Annotations)

**Question 2**

Give few differences between String and StringBuffer classes

String Class is immutable in nature

StringBuffer are mutable objects.

+ operator is used by String to Concantenate

append operator is used by StringBuffer

**Question 3**

What is fail fast iteration in java?

An iterator is considered fail-fast if it throws a ConcurrentModificationException under either of the following two conditions:

1. In multithreaded processing: if one thread is trying to modify a *Collection* while another thread is iterating over it.
2. In single-threaded or in multithreaded processing: if after the creation of the *Iterator*, the container is modified at any time by any method other than the *Iterator*'s own *remove* or *add* methods.

**Question 4**

List as many sub interface(s) for Collection interface in java

JDK 1.4

[BeanContext](http://java.sun.com/j2se/1.4.2/docs/api/java/beans/beancontext/BeanContext.html), [BeanContextServices](http://java.sun.com/j2se/1.4.2/docs/api/java/beans/beancontext/BeanContextServices.html), [List](http://java.sun.com/j2se/1.4.2/docs/api/java/util/List.html), [Set](http://java.sun.com/j2se/1.4.2/docs/api/java/util/Set.html), [SortedSet](http://java.sun.com/j2se/1.4.2/docs/api/java/util/SortedSet.html)

JDK 1.5

[BeanContext](http://java.sun.com/j2se/1.5.0/docs/api/java/beans/beancontext/BeanContext.html), [BeanContextServices](http://java.sun.com/j2se/1.5.0/docs/api/java/beans/beancontext/BeanContextServices.html), [BlockingQueue](http://java.sun.com/j2se/1.5.0/docs/api/java/util/concurrent/BlockingQueue.html)<E>, [List](http://java.sun.com/j2se/1.5.0/docs/api/java/util/List.html)<E>, [Queue](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Queue.html)<E>, [Set](http://java.sun.com/j2se/1.5.0/docs/api/java/util/Set.html)<E>, [SortedSet](http://java.sun.com/j2se/1.5.0/docs/api/java/util/SortedSet.html)<E>

JDK 1.6

[BeanContext](http://java.sun.com/javase/6/docs/api/java/beans/beancontext/BeanContext.html), [BeanContextServices](http://java.sun.com/javase/6/docs/api/java/beans/beancontext/BeanContextServices.html), [BlockingDeque](http://java.sun.com/javase/6/docs/api/java/util/concurrent/BlockingDeque.html)<E>, [BlockingQueue](http://java.sun.com/javase/6/docs/api/java/util/concurrent/BlockingQueue.html)<E>, [Deque](http://java.sun.com/javase/6/docs/api/java/util/Deque.html)<E>, [List](http://java.sun.com/javase/6/docs/api/java/util/List.html)<E>, [NavigableSet](http://java.sun.com/javase/6/docs/api/java/util/NavigableSet.html)<E>, [Queue](http://java.sun.com/javase/6/docs/api/java/util/Queue.html)<E>, [Set](http://java.sun.com/javase/6/docs/api/java/util/Set.html)<E>, [SortedSet](http://java.sun.com/javase/6/docs/api/java/util/SortedSet.html)<E

**Question 5**

Explain dynamic class loading in java.

Loading a class using reflection apis in java

public class MainClass {

public static void main(String[] args){

ClassLoader classLoader = MainClass.class.getClassLoader();

try {

Class aClass = classLoader.loadClass("com.jenkov.MyClass");

System.out.println("aClass.getName() = " + aClass.getName());

} catch (ClassNotFoundException e) {

e.printStackTrace();

}

}

**Question 6**

What is the result of attempting to compile and run the program?

**public** **class** Test {

**public** **static** **void** main(String args[]) **throws** Throwable {

**int** i = 10;

**int** y = 100;

*swap*(i, y);

System.*out*.println("The output is: i = " + i + " y = " + y);

Integer a = **new** Integer(1);

Integer b = **new** Integer(2);

*swap*(a, b);

System.*out*.println("The output is: i = " + a

+ " y = " + b);

}

**public** **static** Integer swap(Integer i, Integer y) **throws** Throwable {

i = y;

**return** i;

}

}

Depends on the JDK that its being run on.

On JDK 1.5 and above : The output is: i = 10 y = 100

The output is: i = 1 y = 2

On JDK 1.4: Program does not compile.

**Question 7**

Are there ways to explicitly request to run finalize method of objects ready for garbage collection and if yes what are the side effects.

Finalize method can be explicitly invoked by calling System.runFinalization() method. But does not guarantee that finalize method will be called. Completely dependent on the jvm to run finalize method.

**Question 8**

What is the output of this program?

**public** **class** TestTwo {

p**ublic** **class** MyException **extends** Error {

}

p**ublic** **class** GivenException **extends** MyException {

}

**public** **class** TestException {

**public** **void** throwException() **throws** Error {

**if** (1 == 1)

**new** MyException();

}

};

**public** **static** **void** main(String args[]) {

**try** {

TestTwo t = **new** TestTwo();

TestException tex = t.**new** TestException();

tex.throwException();

} **catch** (Exception ex) {

System.*out*.println("Caught Exception");

} **catch** (GivenException Myex) {

System.*out*.println("Caught My Exception");

}

}

}

The program exits without printing anything. The Exception must be thrown using throw.

**Question 9**

What is the output of this program

**public** **class** TestThree {

**public** **static** **void** main(String args[]) {

Integer i = **new** Integer(5);

Integer j = **new** Integer(5);

**if** (i == j) {

System.*out*.println("i = j");

}

**if** (i.equals(j)) {

System.*out*.println("i equals j");

}

}

}

Program will print i equals j

**Question 10**

Explain wait and notifyAll in context of threads.

**wait()** method causes the current thread to wait until another thread notifies it of a condition change. You use wait() in conjunction with notify() or notifyAll() to coordinate the activities of multiple threads using the same resources.

**notifyAll( )** wakes up all the threads that called **wait( )** on the same object. The highest priority thread will run first.

**Question 11**

Write an algorithm or pseudo code to reverse a Linked List using recursion.

ReverseLinkeList(List \*\* pList)   
{   
List \* ptr,\*pre = 0, \*next;   
ptr = \*pList;   
next = ptr->next;   
while(ptr->next)   
{   
ptr-> next = pre;   
pre = ptr;   
ptr = next;   
next = next-> next;   
}   
\*pList = ptr;   
}

**Question 12**

Given an array of n elements (containing only positive numbers) and sum, X. Find the first two elements in the array that sum upto X  
  
eg: Array of elements - {2, 3,1000, 200, 51, 88, 29, 49, 65, 40, 98, 12, 3}  
  
Sum - 100. Answer is 51 and 49

initialize a list *S* to contain one element 0.

for each *i* from 1 to *N* do

let *T* be a list consisting of *xi+y*, for all *y* in *S*

let *U* be the union of *T* and *S*

sort *U*

make *S* empty

let *y* be the smallest element of *U*

add *y* to *S*

for each element *z* of *U* in increasing order do //trim the list by eliminating numbers close one to another

if *y<(1-c/N)z*, set *y=z* and add *z* to *S*

if *S* contains a number between *(1-c)s* and *s*, output *yes*, otherwise *no*

**Question 13**

Given Car class can go through multiple safety checks like tyre quality check, engine quality check and brake quality check. Each of these checks can go through independently of each other and update a Boolean quality variable to true or false based on the result. Write a multi threaded program that updates this variable with quality check approved when all the three checks pass. If a thread updates any one check to false other thread should stop working on the car.

**Question 14**

Given a Employee table

Emp\_id First\_Name Last\_Name Manger\_id

100 X Y null

110 A B 100

120 A B 110

130 E F 100

140 G H 120

1. Write a query to get the first Name, Last Name of a employee and his manager
2. Write a query to find all employees with duplicate names.
3. Select a.emp\_id , a.first\_name, a.last\_name, m.emp\_id, m.first\_name, m.last\_name

From Employee a, Employee m where a.manger\_id = m.emp\_id(+)

1. Select e.\* from EMPLOYEE e, (select First\_Name, Last\_Name from EMPLOYEE group by First\_Name, Last\_Name having count (\*) > 1) a where a. First\_Name = e.first\_name and a.last\_name = e.last\_name

**Question 15**

Explain merge sort algorithm with example

**function** merge\_sort(m)

**if** length(m) ≤ 1

**return** m

**var** *list* left, right, result

**var** *integer* middle = length(m) / 2

**for each** x **in** m **up to** middle

add x to left

**for each** x **in** m **after** middle

add x to right

left = merge\_sort(left)

right = merge\_sort(right)

**if** left.last\_item > right.first\_item

result = merge(left, right)

**else**

result = append(left, right)

**return** result

**Question 16**

Give the controller class in Struts framework.

ActionServlet is the controller class in struts

**Question 17**

Give examples of JSP implicit objects.

Application, Config, Exception, Out, Page, Pagecontext, Request, Response and Session

**Question 18**

Explain the differences between ServletConfig and ServletContext.

Parameters defined in the ServletContext are visible for all servlets  
  
Where as parameters defined in the ServletConfig are visible to the servlet where it has been defined .  
  
ServletContext has application scope wide where as ServletConfig has Servlet scope  
  
ServletContext has a life time as that of container.

**Question 19**

Explain various stages in life cycle of jsp.

The generated servlet class for a JSP page implements the HttpJspPage interface of the javax.servlet.jsp package. Hte HttpJspPage interface extends the JspPage interface which inturn extends the Servlet interface of the javax.servlet package. the generated servlet class thus implements all the methods of the these three interfaces. The JspPage interface declares only two mehtods - jspInit() and jspDestroy() that must be implemented by all JSP pages regardless of the client-server protocol. However the JSP specification has provided the HttpJspPage interfaec specifically for the JSp pages serving HTTP requests. This interface declares one method \_jspService().   
The jspInit()- The container calls the jspInit() to initialize te servlet instance.It is called before any other method, and is called only once for a servlet instance.  
The \_jspservice()- The container calls the \_jspservice() for each request, passing it the request and the response objects.  
The jspDestroy()- The container calls this when it decides take the instance out of service. It is the last method called n the servlet instance.

**Question 20**

How can one track session in web application?

1. **Cookies.** You can use HTTP cookies to store information about a shopping session, and each subsequent connection can look up the current session and then extract information about that session from some location on the server machine. This is an excellent alternative, and is the most widely used approach. However, even though servlets have a high-level and easy-to-use interface to cookies, there are still a number of relatively tedious details that need to be handled:
   * Extracting the cookie that stores the session identifier from the other cookies (there may be many, after all),
   * Setting an appropriate expiration time for the cookie (sessions interrupted by 24 hours probably should be reset), and
   * Associating information on the server with the session identifier (there may be far too much information to actually store it in the cookie, plus sensitive data like credit card numbers should *never* go in cookies).
2. **URL Rewriting.** You can append some extra data on the end of each URL that identifies the session, and the server can associate that session identifier with data it has stored about that session. This is also an excellent solution, and even has the advantage that it works with browsers that don't support cookies or where the user has disabled cookies. However, it has most of the same problems as cookies, namely that the server-side program has a lot of straightforward but tedious processing to do. In addition, you have to be very careful that every URL returned to the user (even via indirect means like Location fields in server redirects) has the extra information appended. And, if the user leaves the session and comes back via a bookmark or link, the session information can be lost.
3. **Hidden form fields.** HTML forms have an entry that looks like the following: <INPUT TYPE="HIDDEN" NAME="session" VALUE="...">. This means that, when the form is submitted, the specified name and value are included in the GET or POST data. This can be used to store information about the session. However, it has the major disadvantage that it only works if every page is dynamically generated, since the whole point is that each session has a unique identifier.