**Question 1**

Explain java’s metadata support by providing examples.

JAVA’s metadata support comes with annotations used can be used at class level, attribute level or method level to provide meta information at runtime. User can define their own meta data using ‘@’ before interface

public @interface RequestForEnhancement {

int id();

String synopsis();

String engineer() default "[unassigned]";

String date(); default "[unimplemented]";

}

**Question 2**

Explain the difference between List and Set interfaces in java.

set stores elements in an unordered way but does not contain duplicate elements,where as List stores elements in an ordered way but may contain duplicate elements. Some example of set include HashSet, TreeSet etc. Some example of List include ArrayList, Vector etc.

**Question 3**

Explain Throwable, Error, Exception in java

**Throwable**

The Throwable class is the superclass of all errors and exceptions in the Java language. Only objects that are instances of this class (or one of its subclasses) are thrown by the Java Virtual Machine or can be thrown by the Java throw statement. Similarly, only this class or one of its subclasses can be the argument type in a catch clause.

**Error**

An Error is a subclass of Throwable that indicates serious problems that a reasonable application should not try to catch. Most such errors are abnormal conditions. The ThreadDeath error, though a "normal" condition, is also a subclass of Error because most applications should not try to catch it.

A method is not required to declare in its throws clause any subclasses of Error that might be thrown during the execution of the method but not caught, since these errors are abnormal conditions that should never occur.

**Exception**

The class Exception and its subclasses are a form of Throwable that indicates conditions that a reasonable application might want to catch.

**Question 4**

Differences between Iterator and Enumerator with regards iteration.

Iterator: Fail Safe way of iterating through Collection provides ways to iterate and modify Collection.

Enumerator: Legacy way of iterating through Collection without modifying it.

**Question 5**

Does java use Pass by Value or pass by reference.

Java always uses pass by value in case of objects it’s the value of (or a copy) of memory reference.

**Question 6**

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 7**

Complete the main method to output **“CAB”**

**public** **abstract** **class** A {

**public** String output = "A";

}

**public** **abstract** **class** B **extends** A {

**public** String output = "B";

}

**public** **class** C **extends** B {

**public** String output = "C";

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

// Code goes here

}

}

C a = **new** C();

System.*out*.print(a.output+((A)a).output+((B)a).output);

**Question 8**

What is the output of this program

**public** **class** TestEight {

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

**int** blScore = 1;

**int** whScore = 1;

String color = "BLACK";

**int** i = (color == "BLACK") ? blScore++ : whScore++;

System.*out*.println("blScore:"+blScore++);

System.*out*.println("whScore:"+whScore++);

}

}

**Output:**

blScore:2

whScore:1

**Question 9**

What is the output of this program when you run **java TestNine 1 2**

**public** **class** TestNine {

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

**for** (**int** month = 0; month < args.length; month++) {

**switch** (month) {

**case** 1: System.*out*.print("January");

**case** 2: System.*out*.print("February");

**case** 3: System.*out*.print("March");

**case** 4: System.*out*.print("April");

**case** 5: System.*out*.print("May");

**case** 6: System.*out*.print("June");

**case** 7: System.*out*.print("July");

**case** 8: System.*out*.print("August");

**case** 9: System.*out*.print("September");

**case** 10: System.*out*.print("October");

**case** 11: System.*out*.print("November");

**case** 12: System.*out*.print("December");

}

}

}

}

Output is January February March April May June July August September October November December

**Question 10**

What are Daemon threads in java. How do you make a thread daemon?

Any Java thread can be a *daemon* thread. Daemon threads are service providers for other threads running in the same process as the daemon thread. For example, the HotJava browser uses up to four daemon threads named "Image Fetcher" to fetch images from the file system or network for any thread that needs one. The run() method for a daemon thread is typically an infinite loop that waits for a service request.

When the only remaining threads in a process are daemon threads, the interpreter exits. This makes sense because when only daemon threads remain, there is no other thread for which a daemon thread can provide a service.

To specify that a thread is a daemon thread, call the setDaemon method with the argument true.

**Question 11**

There is a Employee Class with attributes for firstName, lastName, age and salary. Write a program that sorts an ArrayList of employees either alphabetically for the names or numerically for the numbers.

**Question 12**

Explain Quick sort algorithm with example.

**function** quicksort(array)

**var** *list* less, greater

**if** length(array) ≤ 1

**return** array

select and remove a pivot value *pivot* from array

**for each** x **in** array

**if** x ≤ pivot **then** append x to less

**else** append x to greater

**return** concatenate(quicksort(less), pivot, quicksort(greater))

**Question 13**

There is a HR department class that is concerned about an employee hire event, employee fire event and employee vacation event for a employee class. And has to shoot an email to employee group each time these events occur. Employees should also have an opportunity to subscribe or unsubscribe from the email group.

Illustrate your design for solving this problem using UML or pseudo class structures.

**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.

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(+)

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

**Question 15**

Explain with pseudo code the second reader’s writer’s problem.

int readcount, writecount; (initial value = 0)

semaphore mutex 1, mutex 2, mutex 3, w, r ; (initial value = 1)

READER

P(mutex 3);

P(r);

P(mutex 1);

readcount := readcount + 1;

if readcount = 1 then P(w);

V(mutex 1);

V(r);

V(mutex 3);

reading is done

P(mutex 1);

readcount := readcount - 1;

if readcount = 0 then V(w);

V(mutex 1);

WRITER

P(mutex 2);

writecount := writecount + 1;

if writecount = 1 then P(r);

V(mutex 2);

P(w);

writing is performed

V(w);

P(mutex 2);

writecount := writecount - 1;

if writecount = 0 then V(r);

V(mutex 2);

**Question 16**

What are the different types of view tags in Struts

Struts provides three different types of view tags:

Html : supports display of form elements

Bean: used to display objects in the page.

Logic: used to iterate over collections displayed in the page.

**Question 17**

Explain the concept of request forwarding and redirection in J2EE.

**Request Forwarding:**

Request forwarding is used to forward request from one resource to another in the server.

**Request Redirection:**

Request Redirection involves sending the response back to browser with regenerates a new request for server.

**Question 18**

Mention some of jsp implicit objects.

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

**Question 19**

How to handle any exceptions in jsp.

Exceptions is jsp can be handled by using errorPage attribute of page tag

Example:

<%@ page errorPage="ExceptionHandler.jsp" %>

**Question 20**

What is the role of web.xml in J2EE.

Web.xml is deployment descriptor of war based deployment in J2EE. Its hold the url to server resource mapping also.