**Question 1**

Explain the Generics and Enhanced for loop features of JDK 1.5 and above.

**Generics**

When you take an element out of a Collection, you must cast it to the type of element that is stored in the collection. Besides being inconvenient, this is unsafe. The compiler does not check that your cast is the same as the collection's type, so the cast can fail at run time.

Generics provides a way for you to communicate the type of a collection to the compiler, so that it can be checked. Once the compiler knows the element type of the collection, the compiler can check that you have used the collection consistently and can insert the correct casts on values being taken out of the collection.

**Enhanced For Loop**

With the use for Generics java provides an easier way of iterating through collection called enhanced for loop which avoids the use of iterator.

Example:

for (Suit suit : suits)

for (Rank rank : ranks)

sortedDeck.add(new Card(suit, rank));

**Question 2**

How can you synchronize a Map?

Using Collections api [synchronizedMap](http://java.sun.com/j2se/1.4.2/docs/api/java/util/Collections.html#synchronizedMap%28java.util.Map%29)that returns a thread safe map.

**Question 3**

What are chained exceptions in java and how do you handle then?

An application often responds to an exception by throwing another exception. In effect, the first exception *causes* the second exception. It can be very helpful to know when one exception causes another. *Chained Exceptions* help the programmer do this.

The following are the methods and constructors in Throwable that support chained exceptions.

Throwable getCause()

Throwable initCause(Throwable)

Throwable(String, Throwable)

Throwable(Throwable)

**Question 4**

What is the significance of equals and hashCode method.?

Equals method of Object can be overridden to have a deep comparison of sub class implementing it.

hashCode: hashCode methods goes in conjunction with equals. For Object to be considered equal hashCode must return the same value.

**Question 5**

Explain the differences between default and protected access modifiers.

**Default**: Default access modifier is no-modifier. i.e when you do not specify any access modifier explicitly for a method, a variable or a class ( FYI : a top-level class can only be default or public access modifiers) it gets the default access. Default access also means “**package-level**” access. That means a default member can be accessed only inside the same package in which the member is declared.

**Protected**: Protected access modifier is the a little tricky and you can say is a superset of the default access modifier. Protected members are same as the default members as far as the access in the same package is concerned. The difference is that, the protected members are also accessible to the subclasses of the class in which the member is declared which are outside the package in which the parent class is present. But these protected members are “**accessible outside the package only through inheritance**“. i.e you can access a protected member of a class in its subclass present in some other package directly as if the member is present in the subclass itself.

**Question 6**

Can constructor throw Exception in java?

Yes constructors can throw Exception in java. Any programs that creates an instance of the Object must handle it.

**Question7**

What is the purpose of assert keyword used in JDK1.4.x?

An assertion is a statement that tests a condition, somewhat as an *if* statement does. When the condition is violated, it triggers an error so that you know where to revise the assumption. This is a facility to reduce the error chances, especially for those beyond expectations.

**Question 8**

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[]) {

// Add code here

}

}

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

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

**Question 9**

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 10**

What is the output of this program?

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

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

}

**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** (MyException Myex) {

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

}

}

}

Program runs fine without printing anything. Exception must be thrown to be caught.

**Question 11**

Write a program that given a positive integer gives the Fibonacci series upto (equals or less than) that number?

(Formula for Fibonnaci series: Fn = Fn-1+Fn-2 and F0 = 0 and F1 = 1)

public class Example {

public static int Fibonacci(int n) {

if(n == 0 || n==1)

return n;

else

return fibonacci (n-1) + fibonacci (n-2);

}

}

**Question 12**

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 13**

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 14**

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 15**

Explain merge sort algorithm with example.

**function** merge(left,right)

**var** *list* result

**while** length(left) > 0 **and** length(right) > 0

**if** first(left) ≤ first(right)

append first(left) to result

left = rest(left)

**else**

append first(right) to result

right = rest(right)

**end while**

**if** length(left) > 0

append left to result

**else**

append right to result

**return** result

**Question 16**

Compare JSP and Servlets as J2EE technologies.

Servlets are resources that are mostly used as controllers for delegating request for performing some logic.

JSP are in fact a servlets but are mostly used rendering views.

**Question 17**

What is URL Encoding?

URLs can only be sent over the Internet using the [ASCII character-set](http://www.w3schools.com/tags/ref_ascii.asp).

Since URLs often contains characters outside the ASCII set, the URL has to be converted. URL encoding converts the URL into a valid ASCII format.

URL encoding replaces unsafe ASCII characters with "%" followed by two hexadecimal digits corresponding to the character values in the ISO-8859-1 character-set.

URLs cannot contain spaces. URL encoding normally replaces a space with a + sign.

**Question 18**

Give examples of different types of Action classes in struts.

Action, ForwardAction, RedirectAction, DispachAction, MappingDispatchAction and LookUpDispatchAction

**Question 19**

Explain MVC pattern.

**Model:** The model object knows about all the data that need to be displayed. It is model who is aware about all the operations that can be applied to transform that object. It only represents the data of an application. The model represents enterprise data and the business rules that govern access to and updates of this data. Model is not aware about the presentation data and how that data will be displayed to the browser.

**View :** The view represents the presentation of the application. The view object refers to the model. It uses the query methods of the model to obtain the contents and renders it. The view is not dependent on the application logic. It remains same if there is any modification in the business logic. In other words, we can say that it is the responsibility of the of the view's to maintain the consistency in its presentation when the model changes.

**Controller:**  Whenever the user sends a request for something then it always go through the controller. The controller is responsible for intercepting the requests from view and passes it to the model for the appropriate action. After the action has been taken on the data, the controller is responsible for directing the appropriate view to the user. In  GUIs, the views and the controllers often work very closely together.

**Question 20**

Give examples of implicit objects in jsp.

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