**1.关于工作**

1.常用语



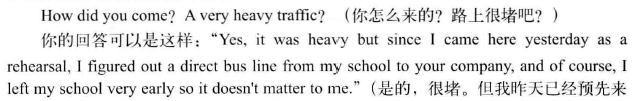




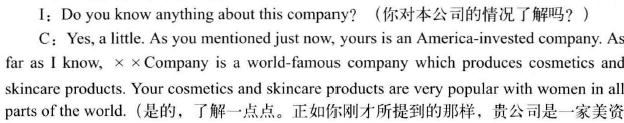




As my previous experience shown.

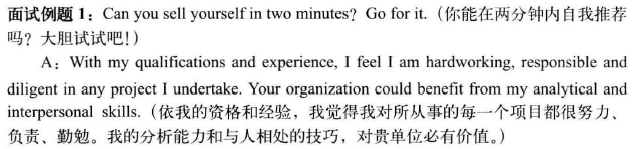
2. 

3.

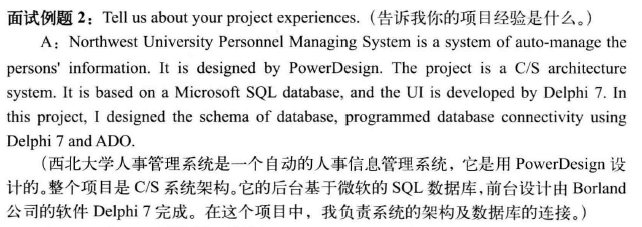


4. 公司文化 规模 业务 未来发展

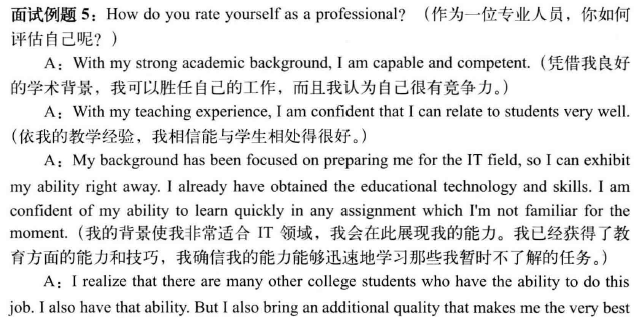
5.

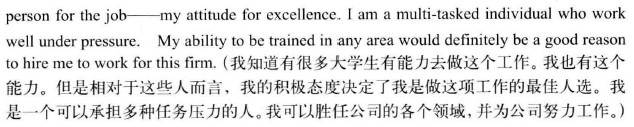


6.

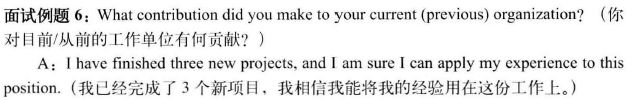


9.

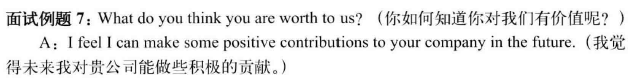




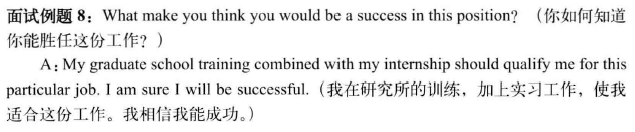
10.



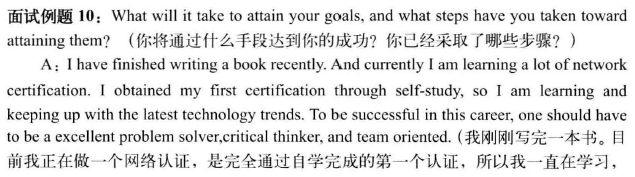
11.

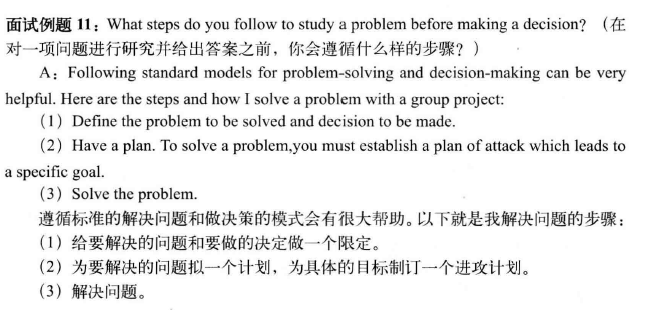


12.

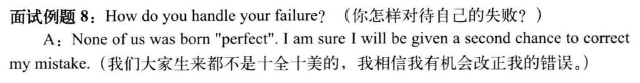


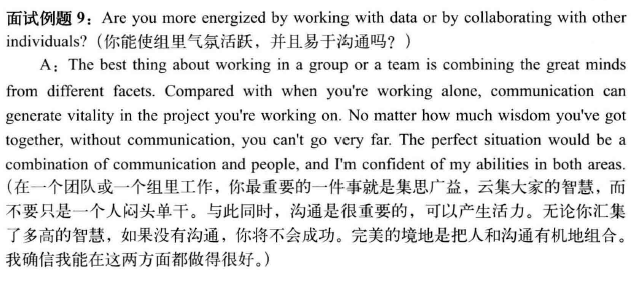
14.

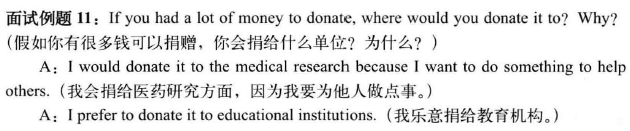




**2 关于个人**

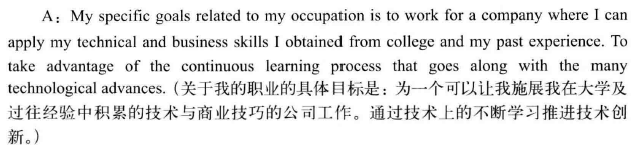


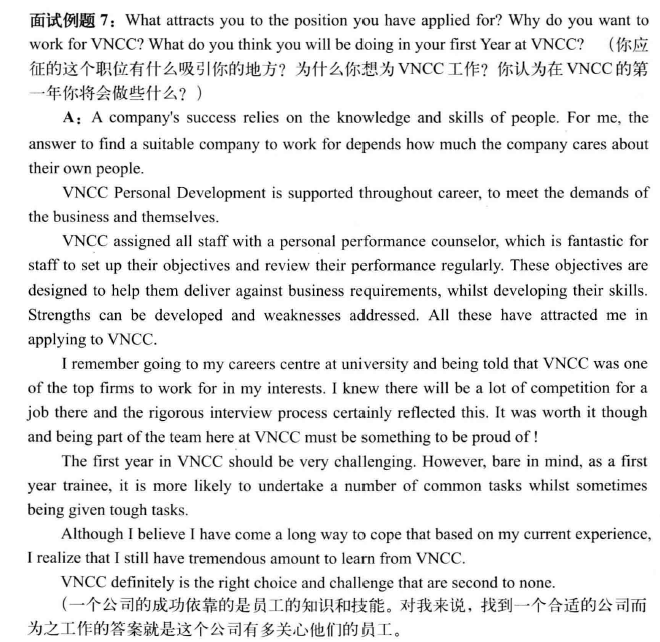


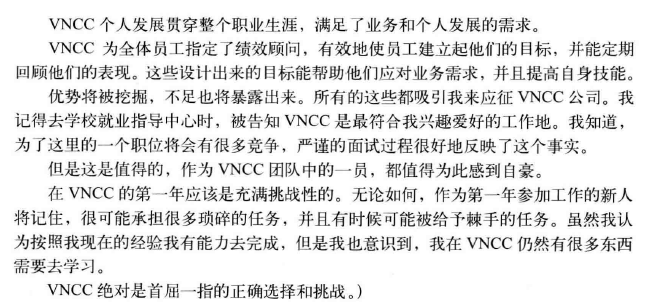


**3.关于未来**









**4. 50题**

**00 介绍**

Good afternoon，everyone！it honoed to have this oputunity to stay here to introduce my self to you. My name is Yutong He, I am 26 years old now . I come form huanggang hubei provice.

7 years ago ，I graduate form Hefei University of Technology， then I got my master’s degree of major named Instrument Science and Technology from ustc. I have been working as a software programmer for one year.

During the past one year , I have been keeping studying to improve my professional skill. I have learned a lot from projects in company. Mainly, I have done two projects, one is a duty system for guards, and the other one is a multifunctional command and dispatch system. Take the second one as an example. I was mainly responsible for writing part codes of the server using java and writing codes of web end using front-end technology. It make me familiar with skills like spring,springmvc,mybatis,mysql,redis,rabbitmq,backnone,require，ztree，jQuery，ligerui and baidumap and so on.

During the procedure of modifying bugs , I have Improved the ability of analyzing and solving problems

Wrote a patent for the company

01 Could you say something about yourself / How would you describe yourself?

|  |
| --- |
| With my experience, I feel I am hardworking, responsible and diligent(勤勉) in any project I undertake.  Your organization could benefit from my professional skill and interpersonal skills. |

A2:说 出 你 的 卖 点

02 Give me a summary of your current job description.

|  |
| --- |
| l have been working as a computer programmer for the past one year. To be specific, I Wrote the summary design document, wrote codes , do functional testing and modified the priogram’s defects. I have been keeping studying for the past one year and have made a lot progress in my Programming skills |

03 Why did you leave your last job?

|  |
| --- |
| Well, I am hoping to get an offer of a better position. I am looking for a company that I really want to work for so that I could settle down and make a long-term contribution  And Autodesk is a great opportunity for me, I hope I can take it. |

04 How do you rate yourself as a professional?

你如何评估自己是位专业人员呢?

|  |
| --- |
| With my previous experience, I think I am capable and competent. |

05 What contribution did you make to your current/previous organization?

A: I have finished three new projects, and I am sure I can apply my experience to this position.

我已经完成三个新项目，我相信我能将我的经验用在这份工作上。

06 Now please tell me something about your achievements in your work unit.

现在请你谈谈你在工作单位的业绩。

|  |
| --- |
|  |

07 Have you received any honors or rewards?

你得到过什么荣誉和奖励吗?

|  |
| --- |
| Won the title of outstanding graduates both of Anhui province and of my university.  Excellent cadres of Graduate Association scholarship |

09 What do you think you are worth to us?

|  |
| --- |
| I feel I can make some positive contributions to your company in the future.  For example |

10 What have you learned from jobs you have held?

12 Would you tell me the essential qualities a programmer should maintain?

|  |
| --- |
| Well, to begin with, I think he need to have a good attitude and good learning ability. and the second point is that he can keep studying to keep up with the trend of technology and Get a good coding ability. |

13 What make you think you would be a success in this position?

should qualify me for this particular job. l am sure l will be successful.

我在大学的训练，加上实习工作，使我适合这份工作。我相信我能成功。

|  |
| --- |
| I am interested in programming and have self-driving force to constantly improve my skills. It should qualify me for this job. l am sure l will be successful. |

14 Are you a multi-tasked individual/ Do you work well under stress or pressure?

你是一位可以同时承担数项工作的人吗/你能承受工作上的压力吗？

A: Yes, I think so.

The feature is needed in my current position, and I know I can handle it well.

15 Did you do any part-time jobs when you studied at your university?

|  |
| --- |
| Yes, I worked as a tutor(家教) for a middle school student. |

16 What is your strongest trait(s)?你个性上最大的特点是什么?

|  |
| --- |
| Helpful to people around me.  Cheerfulness and friendliness.  Working hard and not afraid of hardship |

17 How would your friends or colleagues describe you?

|  |
| --- |
| They say I m an honest, hardworking and responsible person  They feel warm to be together with me. |

18 How would your boss describe you?你的上司怎么评价你?

A: My boss would describe me as hard working, loyal, friendly and committed.

He would say that I work well on my own initiative and deliver what he wants on time and to a high standard.

|  |
| --- |
| My team leader would describe me as hard-working, loyal and reaponsble. He praised that I have made most progress among the developers in our department for many times. |

19 What personality traits do you admire?

你欣赏哪种性格的人?

|  |
| --- |
| I admire a person who is honest, flexible , responsible for his work and easy-going |

21 How do you normally handle criticism?

你通常如何处理别人的批评?

|  |
| --- |
| I can accept constructive criticism. But sometines, if I am not agree with you, I will discuss it later when we cool off. |

22 Q: What do you find frustrating in a work situation?

在工作中，什么事令你不高兴?

|  |
| --- |
| Sometimes, the narrow-minded people make me not happy. Be treated unfairly |

23 How do you deal with those who you think are difficult to work with?

你怎样应付那些你认为难以合作的人?

|  |  |
| --- | --- |
|  |  |

A1: I stick to my principles and keep to rule.

我会坚持原则和谨守规则。

A2: Sometimes, they are just lack of enthusiasm and I got them involved with something constructive. Some of them changed their attitudelater.

有时候，他们只是缺乏工作热情，当我安排他们处理一些建设性的工作时，一些人改变了原来的态度。

24 How do you handle your conflict with your colleagues in your work?

你如何处理与同事在工作中的意见不和?

|  |
| --- |
| l will try to present my ideas in a more clear way in order to make he understand my point of view. |

25 How do you handle your failure?

你怎样对待自己的失败?

|  |
| --- |
| None of us was born “perfect'. I am sure I will be given a second chance to correct my mistake. learn lessons from failure and do better next time. |

26 What provide you with a sense of accomplishment?

|  |
| --- |
| try my best to finish a project and improve my skills during the procedure. Doing my best job for the company. |

27 What is most important in your life right now?

眼下你生活中最重要的是十么?

|  |
| --- |
| To get a job with good prospects for development in my field is most important to me. |

28 What basic principles do you apply to your life?

你在生活中的基本原则是什么?

|  |
| --- |
|  |

29 Tell me what you know about our company.

你对我们公司了解多少?

A: To the best of my knowledge. Your company was originally a small workshop set up in 1978. At that time you had only over ten workers who made some farm tools. After more than ten years of hard work, you have developed into a large enterprise with nearly one thousand technicians and workers. Now your yearly output value has reached as much as over one billion yuan. What a wonder you have done!

30 How long would you like to stay with this company?

你会在公司服务多久?

|  |
| --- |
| I will stay as long as I can continue to learn and to grow in my field. |

31 : Could you project what you would like to be doing five years from now?

你能预料五年后你会做什么吗?

|  |
| --- |
| I hope to show my ability and talents in my field. Perhaps, an opportunity at a management position would be exciting. |

32 What range of pay scale are you interested in?

|  |
| --- |
| The salary range on the job offer. Money is important, but the responsibility that goes along with this job is what interests me the most. |

33 What is your greatest strength?

你的最大长处是十么?

|  |
| --- |
| l suppose a strong point is that  I can take on jobs that bother other people and work at them slowly until they get done. |

34 What are your weak points?

你的缺点是什么?

|  |
| --- |
| A1: When l think something is right, I will stick to that. Sometimes it sounds a little stubborn but l am now trying to find a balance between insistence and compromise.  若我认为某件事是对的， 我会坚持到底。 有时侯，这显得有点顽固，因此我正努力在执着与妥协之间。  As a young girl, I'm a bit shy around people. That is my weakness. But I'm patient both with people and my work.  Well, I'm afraid I'm a poor talker, and that's not very good, so I've been learning how to speak in public. |

35 What are the problems you have encountered in your job?

你在工作中曾遇到过哪些困难?

|  |
| --- |
| At the beginning of coming to work, there were no old staff to guide me. I have to do all the work on my own. |

36 What do you do in your spare time?

你在业余时间如何消遣?

|  |
| --- |
| I have been keeping studying in the past one year. I have many hobbies, I like taking photos and cooking. I also like to riding bikes. |

37 What is your favorite sport?

你喜欢的运动项目是什么?

|  |
| --- |
| I like rading a bike. because I can see the scenery during this process. |

38 How has your college experience prepared you for a business career?

你大学的哪些经历为你的工作打下了基础?

A: l have prepared myself to transition into the workforce through real-world experience involving travel abroad, internship, and entrepreneurial opportunities. While interning with a private organization in Ecuador, I developed a 15-page marketing plan composed in Spanish thatrecommended more effective ways the company could promote its services.

39 How does this job fit into your career plans?

这份工作与你的职业规划有何关联?

A: My ambition is to lead a department and be involved in strategic decision-making at a senior level. l expect the role I'm applying for to give me more experience of leading and some introduction to strategy so that I'm ready for a senior role in a few years.

40 Q: What influenced you to choose this career?

是什么影响你选择这份职业?

A: My past experiences have shown me that I enjoy facing and overcoming the challenge of making a sale. Without a doubt, once l have practiced my presentation and prepared myself for objections, I feel very confident approaching people l don't know and convincing them that they need my product. Lastly, l like sales because my potential for success is limited only by how much of myself I dedicate toward my goal. If any profession is founded on self-determinism, it surely must be sales.

41 What will it take to attain your goals, and what steps have you taken toward attaining them?

为达到目标，你需要做些什么? 以及你已经做了什么？

A: I've already done somere search on other workers at Merrill Lynch to see how they achieved similar goals. I know that Merrill Lynch encourages the pursuit and will reimburse fortuition of a graduate degree. I plan on pursuing a MBA to give me an even more extensive knowledge of business and financial analysis.

42 How did you handle achallenge?

你怎么应对挑战?

When the software development of our new product stalled, I coordinated the team which managed to get the schedule back on track.

We were able to successfully trouble shoot the issues and solve the problems, within a very short period of time.

.

43 What do you look for in a job?

你想要怎样的工作?

I like a job which can make me grow up in my field.

Have a passion for work

Have a good prospect

Have a good team to cooperate.

A responsible leader

44 What is your personal mission statement or motto?

你的座右铬是什么?

A: to Receive what you have to receive, and to change what you can change.

45 What do you enjoy most about what you do now?

你喜欢现在工作的哪部分?

The work of programming is full of challenge, it makes me to keep studying to get more knowledge and improve my skills. I feel vary richness and have a sense of accomplishment.

46 How have you been getting on with your studies so far?

你在校时的成绩怎么样?

A: l have been doing quite well at college. According to the academic records I‘ve achieved so far.

48 Were you in a leading position when you were a college student?

你读大学肘有没有担任过学生干部？

A1: Yes, l was president of Public Relations Association, and I joined the Communist Party of China in my junior year.

49 The English competency is very important here. How is your English proficiency?

你的英语平如何?

A: I have passed CET 6 and BUSINESS ENGLISH CERTIFICATE of middle-lever, I am especially good at reading and listening. If I have the opportunity to enter a Foreign companies. I think I will improve my English a lot.

50 Do you have any questions you would like to ask me?

你有什么问题想问我吗?

A1: when does this position was created?

A3 what are the duties of this position?

A2: I have one question to ask: What have you enjoyed most about working here?

在这工作最开心的是?

A4: What are the challenges that will face the person filling this position?

填补这个职位空缺的人将会面临怎样的挑战?

A6: lf I got the position, can you give me examples of ways I would collaborate with my manager?

如果得到了这个职位，能否举几个例子告诉我如何和我的经理台作?

1. Tell me about yourself。

这是面试问题中的经典头道“开胃菜”，主要考察求职者的个性是否符合公司的企业文化，所以回答时要尽量贴近这个公司的情况来推销自己。比如这个公司属于汽车行业，那么你就该回答：“Driving is my hobby. I really love the feeling of running on the road and I take cars as my best friends。要找到自己跟这个公司的切合点，让面试官了解你适合这个环境。

2. What are your three greatest strengths?

回答这个问题时不可以只是简单回答：“I’m really organized, punctual and get along well with others。”简洁、精炼找到这个公司所看重的特长，是回答这个问题的关键。像要应聘金融行业，可以说：“I think my three strongest strengths are details-oriental, patient and cautious。”但如果应聘者是一个销售精英，可以说：“I think my strongest strengths are aggressive, hard working and communicative。”总之，需要考虑到应聘公司及职务的需要来选择表现自己的优势。

3. What are your three greatest weaknesses?

老板都愿意找到一个了解、热爱工作，并能够为公司创造价值的员工。所以不要很诚实地把自己的毛病都暴露出来，而是要有策略地选择一些能够为自己加分的缺点。

4. Why are you interested in working for our company?

此问题主要考察应聘者是否了解这个工作，或者是否真正有兴趣，所以需要围绕整个公司的具体情况，让面试官知道你很清楚这个公司的运营模式，以及对这项工作非常积极。建议不要以“It seemed like a good career move。”或是“I havent been able to find anything else interesting。”为开始。如果能有一些实际回答的话，加分会更多。应聘者可以这样回答：“I read an article in the newspaper and was very impressed by ……”但是前提是你必须真的对这家公司作了研究，同时也要注意，不要让面试官觉得你是为了讨好面试官而言过其实。

5. Why did you leave your last job?

跳槽的原因可能是因为上一份工作非常糟糕，但面试并不是适合抱怨的场合，而且千万不要对现有或是过去的雇主或同事进行这样的评论：“I didnt agree with the companys direction。”“I got no recognition for my work。”“My boss was totally unreasonable。”铁芸建议，要从自身发展的角度出发来回答：“I love my last job and really learned a lot from it, but I need a new platform and bigger space to make my career successful。”总之，要让面试官知道，以前的公司在你的眼中并非一钱不值，你是怀着感激来评价他们的。

6. If we hire you，how long will you stay with us?

老板想知道你会不会像离开上一份工作那样很快也离开这家公司，你可以回答：“As long as my position here allows me to learn and to advance at a pace with my abilities. ”这样就告诉老板，你是希望同公司一起成长，不会轻易地离开。

应聘者在英语面试的过程中，一定要多说面试官喜欢的英语短语，只有这样，你才能够在英语面试中脱颖而出。

三．Java英语面试题

\* Q1. How could Java classes direct program messages to the system console, but error messages, say to a file?.

The class System has a variable out that represents the standard output, and the variable err that represents the standard error device. By default, they both point at the system console. This how the standard output could be re-directed:

Stream st = new Stream(new FileOutputStream("output.txt")); System.setErr(st); System.setOut(st);

\* Q2. What's the difference between an interface and an abstract class?

A. An abstract class may contain code in method bodies, which is not allowed in an interface. With abstract classes, you have to inherit your class from it and Java does not allow multiple inheritance. On the other hand, you can implement multiple interfaces in your class.

\* Q3. Why would you use a synchronized block vs. synchronized method?

A. Synchronized blocks place locks for shorter periods than synchronized methods.

\* Q4. Explain the usage of the keyword transient?

A. This keyword indicates that the value of this member variable does not have to be serialized with the object.

When the class will be de-serialized, this variable will be initialized with a default value of its data type (i.e. zero for integers).

\* Q5. How can you force garbage collection?

A. You can't force GC, but could request it by calling System.gc().

JVM does not guarantee that GC will be started immediately.

\* Q6. How do you know if an explicit object casting is needed?

A. If you assign a superclass object to a variable of a subclass's data type, you need to do explicit casting. For example:

Object a; Customer b; b = (Customer) a;

When you assign a subclass to a variable having a supeclass type, the casting is performed automatically.

\* Q7. What's the difference between the methods sleep() and wait()

A. The code sleep(1000); puts thread aside for exactly one second. The code wait(1000), causes a wait of up to one second. A thread could stop waiting earlier if it receives the notify() or notifyAll() call. The method wait() is defined in the class Object and the method sleep() is defined in the class Thread.

\* Q8. Can you write a Java class that could be used both as an applet as well as an application?

A. Yes. Add a main() method to the applet.

\* Q9. What's the difference between constructors and other methods?

A. Constructors must have the same name as the class and can not return a value.

They are only called once while regular methods could be called many times.

\* Q10. Can you call one constructor from another if a class has multiple constructors

A. Yes. Use this() syntax.

\* Q11. Explain the usage of Java packages.

A. This is a way to organize files when a project consists of multiple modules. It also helps resolve naming conflicts when different packages have classes with the same names. Packages access level also allows you to protect data from being used by the non-authorized classes.

\* Q12. If a class is located in a package, what do you need to change in the OS environment to be able to use it?

A. You need to add a directory or a jar file that contains the package directories to the CLASSPATH environment variable. Let's say a class Employee belongs to a package com.xyz.hr; and is located in the file c:\dev\com\xyz\hr\Employee.java. In this case, you'd need to add c:\dev to the variable CLASSPATH. If this class contains the method main(), you could test it from a command prompt window as follows:

c:\>java com.xyz.hr.Employee

\* Q13. What's the difference between J2SDK 1.5 and J2SDK 5.0?

A.There's no difference, Sun Microsystems just re-branded this version.

\* Q14. What would you use to compare two String variables - the operator == or the method equals()?

A. I'd use the method equals() to compare the values of the Strings and the == to check if two variables point at the same instance of a String object.

\* Q15. Does it matter in what order catch statements for FileNotFoundException and IOExceptipon are written?

A. Yes, it does. The FileNoFoundException is inherited from the IOException. Exception's subclasses have to be caught first.

Q16. What is more advisable to create a thread, by implementing a Runnable interface or by extending Thread class?(donated in June 2005)

A. Strategically speaking, threads created by implementing Runnable interface are more advisable. If you create a thread by extending a thread class, you cannot extend any other class. If you create a thread by implementing Runnable interface, you save a space for your class to extend another class now or in future.

Q17. An application needs to load a library before it starts to run, how to code?

One option is to use a static block to load a library before anything is called. For example,

class Test {

static {

System.loadLibrary("path-to-library-file");

}

....

}

When you call new Test(), the static block will be called first before any initialization happens. Note that the static block position may matter.

Q18. What is the main difference between Java platform and other platforms?

The Java platform differs from most other platforms in that it's a software-only platform that runs on top of other hardware-based platforms.

The Java platform has two components:

1. The Java Virtual Machine (Java VM)

2. The Java Application Programming Interface (Java API)

Q19. What is the Java Virtual Machine?

The Java Virtual Machine is a software that can be ported onto various hardware-based platforms.

Q20. What is the Java API?

The Java API is a large collection of ready-made software components that provide many useful capabilities, such as graphical user interface (GUI) widgets.

Q21. What is the package?

The package is a Java namespace or part of Java libraries. The Java API is grouped into libraries of related classes and interfaces; these libraries are known as packages.

Q22. What is native code?

The native code is code that after you compile it, the compiled code runs on a specific hardware platform.

Q23. Can main() method be overloaded?

Yes. the main() method is a special method for a program entry. You can overload main() method in any ways. But if you change the signature of the main method, the entry point for the program will be gone.

Q24. What is the serialization?

The serialization is a kind of mechanism that makes a class or a bean persistence by having its properties or fields and state information saved and restored to and from storage.

Q25. How to make a class or a bean serializable?

By implementing either the java.io.Serializable interface, or the java.io.Externalizable interface. As long as one class in a class's inheritance hierarchy implements Serializable or Externalizable, that class is serializable.

Q26. What is J2EE?

J2EE is an environment for developing and deploying enterprise applications. The J2EE platform consists of a set of services, application programming interfaces (APIs), and protocols that provide the functionality for developing multitiered, web-based applications.

Q27. What are the four types of J2EE modules?

1. Application client module

2. Web module

3. Enterprise JavaBeans module

4. Resource adapter module

Q28. What are the differences between Ear, Jar and War files? Under what circumstances should we use each one? (donated in April, 2005)

There are no structural differences between the files; they are all archived using zip-jar compression. However, they are intended for different purposes.

--Jar files (files with a .jar extension) are intended to hold generic libraries of Java classes, resources, auxiliary files, etc.

--War files (files with a .war extension) are intended to contain complete Web applications. In this context, a Web application is defined as a single group of files, classes, resources, .jar files that can be packaged and accessed as one servlet context.

--Ear files (files with a .ear extension) are intended to contain complete enterprise applications. In this context, an enterprise application is defined as a collection of .jar files, resources, classes, and multiple Web applications.

Each type of file (.jar, .war, .ear) is processed uniquely by application servers, servlet containers, EJB containers, etc.

Q29. What two protocols are used in Java RMI technology?

Java Object Serialization and HTTP. The Object Serialization protocol is used to marshal call and return data. The HTTP protocol is used to "POST" a remote method invocation and obtain return data when circumstances warrant.

Q30. Explain for 10 to 15 minutes a technology or product that you are familiar with.(donated in April,2005)

This is to test the person's communication skill and technical skill. If you are really comfortable, start with a public domain technology like Struts or JUnit. If not, sometimes it is better to explain the product that you worked with rather than read about or studied.

Q31: What are different types of inner classes?

A: Nested top-level classes, Member classes, Local classes, Anonymous classes

Nested top-level classes- If you declare a class within a class and specify the static modifier, the compiler treats the class just like any other top-level class.

Any class outside the declaring class accesses the nested class with the declaring class name acting similarly to a package. eg, outer.inner. Top-level inner classes implicitly have access only to static variables.There can also be inner interfaces. All of these are of the nested top-level variety.

Member classes - Member inner classes are just like other member methods and member variables and access to the member class is restricted, just like methods and variables. This means a public member class acts similarly to a nested top-level class. The primary difference between member classes and nested top-level classes is that member classes have access to the specific instance of the enclosing class.

Local classes - Local classes are like local variables, specific to a block of code. Their visibility is only within the block of their declaration. In order for the class to be useful beyond the declaration block, it would need to implement a

more publicly available interface.Because local classes are not members, the modifiers public, protected, private, and static are not usable.

Anonymous classes - Anonymous inner classes extend local inner classes one level further. As anonymous classes have no name, you cannot provide a constructor.

Q32: Are the imports checked for validity at compile time? e.g. will the code containing an import such as java.lang.ABCD compile?

A: Yes the imports are checked for the semantic validity at compile time. The code containing above line of import will not compile. It will throw an error saying,can not resolve symbol

symbol : class ABCD

location: package io

import java.io.ABCD;

Q33: Does importing a package imports the subpackages as well? e.g. Does importing com.MyTest.\* also import com.MyTest.UnitTests.\*?

A: No you will have to import the subpackages explicitly. Importing com.MyTest.\* will import classes in the package MyTest only. It will not import any class in any of it's subpackage.

Q34: What is the difference between declaring a variable and defining a variable?

A: In declaration we just mention the type of the variable and it's name. We do not initialize it. But defining means declaration + initialization.

e.g String s; is just a declaration while String s = new String ("abcd"); Or String s = "abcd"; are both definitions.

Q35: What is the default value of an object reference declared as an instance variable?

A: null unless we define it explicitly.

Q36: Can a top level class be private or protected?

A: No. A top level class can not be private or protected. It can have either "public" or no modifier. If it does not have a modifier it is supposed to have a default access.If a top level class is declared as private the compiler will complain that the "modifier private is not allowed here". This means that a top level class can not be private. Same is the case with protected.

Q37: What type of parameter passing does Java support?

A: In Java the arguments are always passed by value .

Q38: What is the difference between preemptive scheduling and time slicing?

A: Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

Q39. What is ActionServlet?(donated in Nov.2005)

ActionServlet is the Command part of the MVC implementation and is the core of the Struts Framework. ActionServlet creates and uses Action, an ActionForm, and ActionForward. The struts-config.xml file configures the Command. During the creation of the Web project, Action and ActionForm are extended to solve the specific problem space. The file struts-config.xml instructs ActionServlet on how to use the extended classes.

Q40. What is Jakarta Struts Framework? (donated in Nov. 2005)

Jakarta Struts is an open source implementation of MVC (Model-View-Controller) pattern for the development of web based applications. Jakarta Struts is robust architecture and can be used for the development of application of any size. Struts framework makes it much easier to design scalable, reliable Web applications with Java

Q41. What is Struts?

Struts is a web page development framework and an open source software that helps developers build web applications quickly and easily. Struts combines Java Servlets, Java Server Pages, custom tags, and message resources into a unified framework. It is a cooperative, synergistic platform, suitable for development teams, independent developers, and everyone between.

Q42. How is the MVC design pattern used in Struts framework?

In the MVC design pattern, application flow is mediated by a central Controller. The Controller delegates requests to an appropriate handler. The handlers are tied to a Model, and each handler acts as an adapter between the request and the Model. The Model represents, or encapsulates, an application's business logic or state. Control is usually then forwarded back through the Controller to the appropriate View. The forwarding can be determined by consulting a set of mappings, usually loaded from a database or configuration file. This provides a loose coupling between the View and Model, which can make an application significantly easier to create and maintain.

Controller--Servlet controller which supplied by Struts itself; View --- what you can see on the screen, a JSP page and presentation components; Model --- System state and a business logic JavaBeans.

Q43. What are the core classes of Struts?

Action, ActionForm, ActionServlet, ActionMapping, ActionForward are basic classes of Structs.

Question:

Q44. What is the design role played by Struts?

The role played by Structs is controller in Model/View/Controller(MVC) style. The View is played by JSP and Model is played by JDBC or generic data source classes. The Struts controller is a set of programmable components that allow developers to define exactly how the application interacts with the user.

Q45. How Struts control data flow?

Struts implements the MVC/Layers pattern through the use of ActionForwards and ActionMappings to keep control-flow decisions out of presentation layer.

Q46. What configuration files are used in Struts?

--ApplicationResourcesl.properties

--struts-config.xml

These two files are used to bridge the gap between the Controller and the Model.

Q47. What helpers in the form of JSP pages are provided in Struts framework?

--struts-html.tld

--struts-bean.tld

--struts-logic.tld

Q48. What is Servlet?

A servlet is a Java technology-based Web component, managed by a container called servlet container or servlet engine, that generates dynamic content and interacts with web clients via a request\/response paradigm.

Q49. Why is Servlet so popular?

Because servlets are platform-independent Java classes that are compiled to platform-neutral byte code that can be loaded dynamically into and run by a Java technology-enabled Web server.

Q50. What is servlet container?

The servlet container is a part of a Web server or application server that provides the network services over which requests and responses are sent, decodes MIME-based requests, and formats MIME-based responses. A servlet container also contains and manages servlets through their lifecycle.

Q51. When a client request is sent to the servlet container, how does the container choose which servlet to invoke?

The servlet container determines which servlet to invoke based on the configuration of its servlets, and calls it with objects representing the request and response.

Q52. If a servlet is not properly initialized, what exception may be thrown?

During initialization or service of a request, the servlet instance can throw an UnavailableException or a ServletException.

Q53. Given the request path below, which are context path, servlet path and path info?

/bookstore/education/index.html

context path: /bookstore

servlet path: /education

path info: /index.html

Q54. What is filter? Can filter be used as request or response?

A filter is a reusable piece of code that can transform the content of HTTP requests,responses, and header information.

Filters do not generally create a response or respond to a request as servlets do, rather they modify or adapt the requests for a resource, and modify or adapt responses from a resource.

Q55. When using servlets to build the HTML, you build a DOCTYPE line, why do you do that?

I know all major browsers ignore it even though the HTML 3.2 and 4.0 specifications require it. But building a DOCTYPE line tells HTML validators which version of HTML you are using so they know which specification to check your document against. These validators are valuable debugging services, helping you catch HTML syntax errors.

Q56. How do you call a bean method from a Javascript onclick handler? (donated in Oct,2005)

This is a trick question. JavaScript onclick handler is usually handled on the client side and a bean method is handled on the server side. It is not possible to call a bean method from a Javascript onclick handler directly. Explain it why or ask back to clarify this question.

Q57. What is a JSP and what is it used for?

Java Server Pages (JSP) is a platform independent presentation layer technology that comes with SUN s J2EE platform. JSPs are normal HTML pages with Java code pieces embedded in them. JSP pages are saved to \*.jsp files. A JSP compiler is used in the background to generate a Servlet from the JSP page.

Q58. What is JSP technology?

Java Server Page is a standard Java extension that is defined on top of the servlet Extensions. The goal of JSP is the simplified creation and management of dynamic Web pages. JSPs are secure, platform-independent, and best of all, make use of Java as a server-side scripting language.

Q59. What is JSP page?

A JSP page is a text-based document that contains two types of text: static template data, which can be expressed in any text-based format such as HTML, SVG, WML, and XML, and JSP elements, which construct dynamic content.

Q60. What are the implicit objects?

Implicit objects are objects that are created by the web container and contain information related to a particular request, page, or application. They are:

--request

--response

--pageContext

--session

--application

--out

--config

--page

--exception

Q61: What is the difference between an Interface and an Abstract class?

A: An abstract class can have instance methods that implement a default behavior. An Interface can only declare constants and instance methods, but cannot implement default behavior and all methods are implicitly abstract. An interface has all public members and no implementation. An abstract class is a class which may have the usual flavors of class members (private, protected, etc.), but has some abstract methods.

Q62: What is the purpose of garbage collection in Java, and when is it used?

A: The purpose of garbage collection is to identify and discard objects that are no longer needed by a program so that their resources can be reclaimed and reused. A Java object is subject to garbage collection when it becomes unreachable to the program in which it is used.

Q63: Describe synchronization in respect to multithreading.

A: With respect to multithreading, synchronization is the capability to control the access of multiple threads to shared resources. Without synchonization, it is possible for one thread to modify a shared variable while another thread is in the process of using or updating same shared variable. This usually leads to significant errors.

Q64: Explain different way of using thread?

A: The thread could be implemented by using runnable interface or by inheriting from the Thread class. The former is more advantageous, 'cause when you are going for multiple inheritance..the only interface can help.

Q65: What are pass by reference and passby value?

A: Pass By Reference means the passing the address itself rather than passing the value. Passby Value means passing a copy of the value to be passed.

Q66: What is HashMap and Map?

A: Map is Interface and Hashmap is class that implements that.

Q67: Difference between HashMap and HashTable?

A: The HashMap class is roughly equivalent to Hashtable, except that it is unsynchronized and permits nulls. (HashMap allows null values as key and value whereas Hashtable doesnt allow). HashMap does not guarantee that the order of the map will remain constant over time. HashMap is unsynchronized and Hashtable is synchronized.

Q68: Difference between Vector and ArrayList?

A: Vector is synchronized whereas arraylist is not.

Q69: Difference between Swing and Awt?

A: AWT are heavy-weight componenets. Swings are light-weight components. Hence swing works faster than AWT.

Q70: What is the difference between a constructor and a method?

A: A constructor is a member function of a class that is used to create objects of that class. It has the same name as the class itself, has no return type, and is invoked using the new operator.

A method is an ordinary member function of a class. It has its own name, a return type (which may be void), and is invoked using the dot operator.

Q71: What is an Iterator?

A: Some of the collection classes provide traversal of their contents via a java.util.Iterator interface. This interface allows you to walk through a collection of objects, operating on each object in turn. Remember when using Iterators that they contain a snapshot of the collection at the time the Iterator was obtained; generally it is not advisable to modify the collection itself while traversing an Iterator.

Q72: State the significance of public, private, protected, default modifiers both singly and in combination and state the effect of package relationships on declared items qualified by these modifiers.

A: public : Public class is visible in other packages, field is visible everywhere (class must be public too)

private : Private variables or methods may be used only by an instance of the same class that declares the variable or method, A private feature may only be accessed by the class that owns the feature.

protected : Is available to all classes in the same package and also available to all subclasses of the class that owns the protected feature.This access is provided even to subclasses that reside in a different package from the class that owns the protected feature.

default :What you get by default ie, without any access modifier (ie, public private or protected).It means that it is visible to all within a particular package.

Q73: What is an abstract class?

A: Abstract class must be extended/subclassed (to be useful). It serves as a template. A class that is abstract may not be instantiated (ie, you may not call its constructor), abstract class may contain static data. Any class with an abstract method is automatically abstract itself, and must be declared as such.

A class may be declared abstract even if it has no abstract methods. This prevents it from being instantiated.

Q74: What is static in java?

A: Static means one per class, not one for each object no matter how many instance of a class might exist. This means that you can use them without creating an instance of a class.Static methods are implicitly final, because overriding is done based on the type of the object, and static methods are attached to a class, not an object. A static method in a superclass can be shadowed by another static method in a subclass, as long as the original method was not declared final. However, you can't override a static method with a nonstatic method. In other words, you can't change a static method into an instance method in a subclass.

Q75: What is final?

A: A final class can't be extended ie., final class may not be subclassed. A final method can't be overridden when its class is inherited. You can't change value of a final variable (is a constant).

Q76: What if the main method is declared as private?

A: The program compiles properly but at runtime it will give "Main method not public." message.

Q77: What if the static modifier is removed from the signature of the main method?

A: Program compiles. But at runtime throws an error "NoSuchMethodError".

Q78: What if I write static public void instead of public static void?

A: Program compiles and runs properly.

Q79: What if I do not provide the String array as the argument to the method?

A: Program compiles but throws a runtime error "NoSuchMethodError".

Q80: What is the first argument of the String array in main method?

A: The String array is empty. It does not have any element. This is unlike C/C++ where the first element by default is the program name.

Q81: If I do not provide any arguments on the command line, then the String array of Main method will be empty or null?

A: It is empty. But not null.

Q82: How can one prove that the array is not null but empty using one line of code?

A: Print args.length. It will print 0. That means it is empty. But if it would have been null then it would have thrown a NullPointerException on attempting to print args.length.

Q83: What environment variables do I need to set on my machine in order to be able to run Java programs?

A: CLASSPATH and PATH are the two variables.

Q84: Can an application have multiple classes having main method?

A: Yes it is possible. While starting the application we mention the class name to be run. The JVM will look for the Main method only in the class whose name you have mentioned. Hence there is not conflict amongst the multiple classes having main method.

Q85: Can I have multiple main methods in the same class?

A: No the program fails to compile. The compiler says that the main method is already defined in the class.

Q86: Do I need to import java.lang package any time? Why ?

A: No. It is by default loaded internally by the JVM.

Q87: Can I import same package/class twice? Will the JVM load the package twice at runtime?

A: One can import the same package or same class multiple times. Neither compiler nor JVM complains abt it. And the JVM will internally load the class only once no matter how many times you import the same class.

Q88: What are Checked and UnChecked Exception?

A: A checked exception is some subclass of Exception (or Exception itself), excluding class RuntimeException and its subclasses.

Making an exception checked forces client programmers to deal with the possibility that the exception will be thrown. eg, IOException thrown by java.io.FileInputStream's read() method•

Unchecked exceptions are RuntimeException and any of its subclasses. Class Error and its subclasses also are unchecked. With an unchecked exception, however, the compiler doesn't force client programmers either to catch the

exception or declare it in a throws clause. In fact, client programmers may not even know that the exception could be thrown. eg, StringIndexOutOfBoundsException thrown by String's charAt() method• Checked exceptions must be caught at compile time. Runtime exceptions do not need to be. Errors often cannot be.

Q89: What is Overriding?

A: When a class defines a method using the same name, return type, and arguments as a method in its superclass, the method in the class overrides the method in the superclass.

When the method is invoked for an object of the class, it is the new definition of the method that is called, and not the method definition from superclass. Methods may be overridden to be more public, not more private.

Q90: Primitive data types are passed by reference or pass by value?

A: Primitive data types are passed by value.

Q91: Objects are passed by value or by reference?

A: Java only supports pass by value. With objects, the object reference itself is passed by value and so both the original reference and parameter copy both refer to the same object .

Q92: What is serialization?

A: Serialization is a mechanism by which you can save the state of an object by converting it to a byte stream.

Q93: How do I serialize an object to a file?

A: The class whose instances are to be serialized should implement an interface Serializable. Then you pass the instance to the ObjectOutputStream which is connected to a fileoutputstream. This will save the object to a file.

Q94: Which methods of Serializable interface should I implement?

A: The serializable interface is an empty interface, it does not contain any methods. So we do not implement any methods.

Q95: How can I customize the seralization process? i.e. how can one have a control over the serialization process?

A: Yes it is possible to have control over serialization process. The class should implement Externalizable interface. This interface contains two methods namely readExternal and writeExternal. You should implement these methods and write the logic for customizing the serialization process.

Q96: What is the common usage of serialization?

A: Whenever an object is to be sent over the network, objects need to be serialized. Moreover if the state of an object is to be saved, objects need to be serilazed.

Q97: What is Externalizable interface?

A: Externalizable is an interface which contains two methods readExternal and writeExternal. These methods give you a control over the serialization mechanism. Thus if your class implements this interface, you can customize the serialization process by implementing these methods.

Q98: When you serialize an object, what happens to the object references included in the object?

A: The serialization mechanism generates an object graph for serialization. Thus it determines whether the included object references are serializable or not. This is a recursive process. Thus when an object is serialized, all the included objects are also serialized alongwith the original obect.

Q99: What one should take care of while serializing the object?

A: One should make sure that all the included objects are also serializable. If any of the objects is not serializable then it throws a NotSerializableException.

Q100: What happens to the static fields of a class during serialization?

A: There are three exceptions in which serialization doesnot necessarily read and write to the stream. These are

1. Serialization ignores static fields, because they are not part of ay particular state state.

2. Base class fields are only hendled if the base class itself is serializable.

3. Transient fields.

Q101: Does Java provide any construct to find out the size of an object?

A: No there is not sizeof operator in Java. So there is not direct way to determine the size of an object directly in Java.

Q102: Give a simplest way to find out the time a method takes for execution without using any profiling tool?

A: Read the system time just before the method is invoked and immediately after method returns. Take the time difference, which will give you the time taken by a method for execution.

To put it in code...

long start = System.currentTimeMillis ();

method ();

long end = System.currentTimeMillis ();

System.out.println ("Time taken for execution is " + (end - start));

Remember that if the time taken for execution is too small, it might show that it is taking zero milliseconds for execution. Try it on a method which is big enough, in the sense the one which is doing considerable amout of processing.

Q103: What are wrapper classes?

A: Java provides specialized classes corresponding to each of the primitive data types. These are called wrapper classes. They are e.g. Integer, Character, Double etc.

Q104: Why do we need wrapper classes?

A: It is sometimes easier to deal with primitives as objects. Moreover most of the collection classes store objects and not primitive data types. And also the wrapper classes provide many utility methods also. Because of these resons we need wrapper classes. And since we create instances of these classes we can store them in any of the collection classes and pass them around as a collection. Also we can pass them around as method parameters where a method expects an object.

Q105: What are checked exceptions?

A: Checked exception are those which the Java compiler forces you to catch. e.g. IOException are checked Exceptions.

Q106: What are runtime exceptions?

A: Runtime exceptions are those exceptions that are thrown at runtime because of either wrong input data or because of wrong business logic etc. These are not checked by the compiler at compile time.

Q107: What is the difference between error and an exception?

A: An error is an irrecoverable condition occurring at runtime. Such as OutOfMemory error. These JVM errors and you can not repair them at runtime. While exceptions are conditions that occur because of bad input etc. e.g. FileNotFoundException will be thrown if the specified file does not exist. Or a NullPointerException will take place if you try using a null reference. In most of the cases it is possible to recover from an exception (probably by giving user a feedback for entering proper values etc.).

Q108: How to create custom exceptions?

A: Your class should extend class Exception, or some more specific type thereof.

Q109: If I want an object of my class to be thrown as an exception object, what should I do?

A: The class should extend from Exception class. Or you can extend your class from some more precise exception type also.

Q110: If my class already extends from some other class what should I do if I want an instance of my class to be thrown as an exception object?

A: One can not do anytihng in this scenarion. Because Java does not allow multiple inheritance and does not provide any exception interface as well.

Q: How does an exception permeate through the code?

A: An unhandled exception moves up the method stack in search of a matching When an exception is thrown from a code which is wrapped in a try block followed by one or more catch blocks, a search is made for matching catch block. If a matching type is found then that block will be invoked. If a matching type is not found then the exception moves up the method stack and reaches the caller method. Same procedure is repeated if the caller method is included in a try catch block. This process continues until a catch block handling the appropriate type of exception is found. If it does not find such a block then finally the program terminates.

Q111: What are the different ways to handle exceptions?

A: There are two ways to handle exceptions,

1. By wrapping the desired code in a try block followed by a catch block to catch the exceptions. and

2. List the desired exceptions in the throws clause of the method and let the caller of the method hadle those exceptions.

Q112: What is the basic difference between the 2 approaches to exception handling.

1> try catch block and

2> specifying the candidate exceptions in the throws clause?

When should you use which approach?

A: In the first approach as a programmer of the method, you urself are dealing with the exception. This is fine if you are in a best position to decide should be done in case of an exception. Whereas if it is not the responsibility of the method to deal with it's own exceptions, then do not use this approach. In this case use the second approach. In the second approach we are forcing the caller of the method to catch the exceptions, that the method is likely to throw. This is often the approach library creators use. They list the exception in the throws clause and we must catch them. You will find the same approach throughout the java libraries we use.

Q113: Is it necessary that each try block must be followed by a catch block?

A: It is not necessary that each try block must be followed by a catch block. It should be followed by either a catch block OR a finally block. And whatever exceptions are likely to be thrown should be declared in the throws clause of the method.

Q114: If I write return at the end of the try block, will the finally block still execute?

A: Yes even if you write return as the last statement in the try block and no exception occurs, the finally block will execute. The finally block will execute and then the control return.

Q115: If I write System.exit (0); at the end of the try block, will the finally block still execute?A: No in this case the finally block will not execute because when you say System.exit (0); the control immediately goes out of the program, and thus finally never executes.

Q116: How are Observer and Observable used?

A: Objects that subclass the Observable class maintain a list of observers. When an Observable object is updated it invokes the update() method of each of its observers to notify the observers that it has changed state. The Observer interface is implemented by objects that observe Observable objects.

Q117: What is synchronization and why is it important?

A: With respect to multithreading, synchronization is the capability to control

the access of multiple threads to shared resources. Without synchronization, it is possible for one thread to modify a shared object while another thread is in the process of using or updating that object's value. This often leads to significant errors.

Q118: How does Java handle integer overflows and underflows?

A: It uses those low order bytes of the result that can fit into the size of the type allowed by the operation.

Q119: Does garbage collection guarantee that a program will not run out of memory?

A: Garbage collection does not guarantee that a program will not run out of memory. It is possible for programs to use up memory resources faster than they are garbage collected. It is also possible for programs to create objects that are not subject to garbage collection

Q120: What is the difference between preemptive scheduling and time slicing?

A: Under preemptive scheduling, the highest priority task executes until it enters the waiting or dead states or a higher priority task comes into existence. Under time slicing, a task executes for a predefined slice of time and then reenters the pool of ready tasks. The scheduler then determines which task should execute next, based on priority and other factors.

Q121: When a thread is created and started, what is its initial state?

A: A thread is in the ready state after it has been created and started.

Q122: What is the purpose of finalization?

A: The purpose of finalization is to give an unreachable object the opportunity to perform any cleanup processing before the object is garbage collected.

Q123: What is the Locale class?

A: The Locale class is used to tailor program output to the conventions of a particular geographic, political, or cultural region.

Q124: What is the difference between a while statement and a do statement?

A: A while statement checks at the beginning of a loop to see whether the next loop iteration should occur. A do statement checks at the end of a loop to see whether the next iteration of a loop should occur. The do statement will always execute the body of a loop at least once.

Q125: What is the difference between static and non-static variables?

A: A static variable is associated with the class as a whole rather than with specific instances of a class. Non-static variables take on unique values with each object instance.

Q126: How are this() and super() used with constructors?

A: This() is used to invoke a constructor of the same class. super() is used to invoke a superclass constructor.

Q127: What are synchronized methods and synchronized statements?

A: Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

Q128: What is daemon thread and which method is used to create the daemon thread?

A: 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.

Q129: Can applets communicate with each other?

A: At this point in time applets may communicate with other applets running in the same virtual machine. If the applets are of the same class, they can communicate via shared static variables. If the applets are of different classes, then each will need a reference to the same class with static variables. In any case the basic idea is to pass the information back and forth through a static variable.

An applet can also get references to all other applets on the same page using the getApplets() method of java.applet.AppletContext. Once you get the reference to an applet, you can communicate with it by using its public members.

It is conceivable to have applets in different virtual machines that talk to a server somewhere on the Internet and store any data that needs to be serialized there. Then, when another applet needs this data, it could connect to this same server. Implementing this is non-trivial.

Q130: What are the steps in the JDBC connection?

A: While making a JDBC connection we go through the following steps :

Step 1 : Register the database driver by using :

Class.forName(\" driver classs for that specific database\" );

Step 2 : Now create a database connection using :

Connection con = DriverManager.getConnection(url,username,password);

Step 3: Now Create a query using :

Statement stmt = Connection.Statement(\"select \* from TABLE NAME\");

Step 4 : Exceute the query :

stmt.exceuteUpdate();

Q131: How does a try statement determine which catch clause should be used to handle an exception?

A: When an exception is thrown within the body of a try statement, the catch clauses of the try statement are examined in the order in which they appear. The first catch clause that is capable of handling the exceptionis executed. The remaining catch clauses are ignored.

Q132: Can an unreachable object become reachable again?

A: An unreachable object may become reachable again. This can happen when the object's finalize() method is invoked and the object performs an operation which causes it to become accessible to reachable objects.

Q133: What method must be implemented by all threads?

A: All tasks must implement the run() method, whether they are a subclass of Thread or implement the Runnable interface.

Q134: What are synchronized methods and synchronized statements?

A: Synchronized methods are methods that are used to control access to an object. A thread only executes a synchronized method after it has acquired the lock for the method's object or class. Synchronized statements are similar to synchronized methods. A synchronized statement can only be executed after a thread has acquired the lock for the object or class referenced in the synchronized statement.

Q135: What is Externalizable?

A: Externalizable is an Interface that extends Serializable Interface. And sends data into Streams in Compressed Format. It has two methods, writeExternal(ObjectOuput out) and readExternal(ObjectInput in)

Q136: What modifiers are allowed for methods in an Interface?

A: Only public and abstract modifiers are allowed for methods in interfaces.

Q137: What are some alternatives to inheritance?

A: Delegation is an alternative to inheritance. Delegation means that you include an instance of another class as an instance variable, and forward messages to the instance. It is often safer than inheritance because it forces you to think about each message you forward, because the instance is of a known class, rather than a new class, and because it doesn't force you to accept all the methods of the super class: you can provide only the methods that really make sense. On the other hand, it makes you write more code, and it is harder to re-use (because it is not a subclass).

Q138: What does it mean that a method or field is "static"?

A: Static variables and methods are instantiated only once per class. In other words they are class variables, not instance variables. If you change the value of a static variable in a particular object, the value of that variable changes for all instances of that class.

Static methods can be referenced with the name of the class rather than the name of a particular object of the class (though that works too). That's how library methods like System.out.println() work out is a static field in the java.lang.System class.

Q139: What is the catch or declare rule for method declarations?

A: If a checked exception may be thrown within the body of a method, the method must either catch the exception or declare it in its throws clause.

Q140: Is Empty .java file a valid source file?

A: Yes, an empty .java file is a perfectly valid source file.

Q141: Can a .java file contain more than one java classes?

A: Yes, a .java file contain more than one java classes, provided at the most one of them is a public class.

Q142: Is String a primitive data type in Java?

A: No String is not a primitive data type in Java, even though it is one of the most extensively used object. Strings in Java are instances of String class defined in java.lang package.

Q143: Is main a keyword in Java?

A: No, main is not a keyword in Java.

Q144: Is next a keyword in Java?

A: No, next is not a keyword.

Q145: Is delete a keyword in Java?

A: No, delete is not a keyword in Java. Java does not make use of explicit destructors the way C++ does.

Q146: Is exit a keyword in Java?

A: No. To exit a program explicitly you use exit method in System object.

Q147: What happens if you dont initialize an instance variable of any of the primitive types in Java?

A: Java by default initializes it to the default value for that primitive type. Thus an int will be initialized to 0, a boolean will be initialized to false.

Q148: What will be the initial value of an object reference which is defined as an instance variable?

A: The object references are all initialized to null in Java. However in order to do anything useful with these references, you must set them to a valid object, else you will get NullPointerExceptions everywhere you try to use such default initialized references.

Q149: What are the different scopes for Java variables?

A: The scope of a Java variable is determined by the context in which the variable is declared. Thus a java variable can have one of the three scopes at any given point in time.

1. Instance : - These are typical object level variables, they are initialized to default values at the time of creation of object, and remain accessible as long as the object accessible.

2. Local : - These are the variables that are defined within a method. They remain accessbile only during the course of method excecution. When the method finishes execution, these variables fall out of scope.

3. Static: - These are the class level variables. They are initialized when the class is loaded in JVM for the first time and remain there as long as the class remains loaded. They are not tied to any particular object instance.

Q150: What is the default value of the local variables?

A: The local variables are not initialized to any default value, neither primitives nor object references. If you try to use these variables without initializing them explicitly, the java compiler will not compile the code. It will complain abt the local varaible not being initilized..

Q151: How many objects are created in the following piece of code?

MyClass c1, c2, c3;

c1 = new MyClass ();

c3 = new MyClass ();

A: Only 2 objects are created, c1 and c3. The reference c2 is only declared and not initialized.

Q152: Can a public class MyClass be defined in a source file named YourClass.java?

A: No the source file name, if it contains a public class, must be the same as the public class name itself with a .java extension.

Q153: Can main method be declared final?

A: Yes, the main method can be declared final, in addition to being public static.

Q154: What will be the output of the following statement?

System.out.println ("1" + 3);

A: It will print 13.

Q155: What will be the default values of all the elements of an array defined as an instance variable?

A: If the array is an array of primitive types, then all the elements of the array will be initialized to the default value corresponding to that primitive type. e.g. All the elements of an array of int will be initialized to 0, while that of boolean type will be initialized to false. Whereas if the array is an array of references (of any type), all the elements will be initialized to null.

Question156: What is transient variable?

Answer: Transient variable can't be serialize. For example if a variable is declared as transient in a Serializable class and the class is written to an ObjectStream, the value of the variable can't be written to the stream instead when the class is retrieved from the ObjectStream the value of the variable becomes null.

Question: Name the containers which uses Border Layout as their default layout?

Answer: Containers which uses Border Layout as their default are: window, Frame and Dialog classes.

Question157: What do you understand by Synchronization?

Answer: Synchronization is a process of controlling the access of shared resources by the multiple threads in such a manner that only one thread can access one resource at a time. In non synchronized multithreaded application, it is possible for one thread to modify a shared object while another thread is in the process of using or updating the object's value. Synchronization prevents such type of data corruption.

E.g. Synchronizing a function:

public synchronized void Method1 () {

// Appropriate method-related code.

}

E.g. Synchronizing a block of code inside a function:

public myFunction (){

synchronized (this) {

// Synchronized code here.

}

}

Question158: What is Collection API?

Answer: The Collection API is a set of classes and interfaces that support operation on collections of objects. These classes and interfaces are more flexible, more powerful, and more regular than the vectors, arrays, and hashtables if effectively replaces.

Example of classes: HashSet, HashMap, ArrayList, LinkedList, TreeSet and TreeMap.

Example of interfaces: Collection, Set, List and Map.

Question159: Is Iterator a Class or Interface? What is its use?

Answer: Iterator is an interface which is used to step through the elements of a Collection.

Question160: What is similarities/difference between an Abstract class and Interface?

Answer: Differences are as follows:

Interfaces provide a form of multiple inheritance. A class can extend only one other class.

Interfaces are limited to public methods and constants with no implementation. Abstract classes can have a partial implementation, protected parts, static methods, etc.

A Class may implement several interfaces. But in case of abstract class, a class may extend only one abstract class.

Interfaces are slow as it requires extra indirection to find corresponding method in the actual class. Abstract classes are fast.

Similarities:

Neither Abstract classes or Interface can be instantiated.

Question161: How to define an Abstract class?

Answer: A class containing abstract method is called Abstract class. An Abstract class can't be instantiated.

Example of Abstract class:

abstract class testAbstractClass {

protected String myString;

public String getMyString() {

return myString;

}

public abstract string anyAbstractFunction();

}

Question162: How to define an Interface?

Answer: In Java Interface defines the methods but does not implement them. Interface can include constants. A class that implements the interfaces is bound to implement all the methods defined in Interface.

Emaple of Interface:

public interface sampleInterface {

public void functionOne();

public long CONSTANT\_ONE = 1000;

}

Question163: Explain the user defined Exceptions?

Answer: User defined Exceptions are the separate Exception classes defined by the user for specific purposed. An user defined can created by simply sub-classing it to the Exception class. This allows custom exceptions to be generated (using throw) and caught in the same way as normal exceptions.

Example:

class myCustomException extends Exception {

// The class simply has to exist to be an exception

}

Question164: Explain the new Features of JDBC 2.0 Core API?

Answer: The JDBC 2.0 API includes the complete JDBC API, which includes both core and Optional Package API, and provides inductrial-strength database computing capabilities.

New Features in JDBC 2.0 Core API:

Scrollable result sets- using new methods in the ResultSet interface allows programmatically move the to particular row or to a position relative to its current position

JDBC 2.0 Core API provides the Batch Updates functionality to the java applications.

Java applications can now use the ResultSet.updateXXX methods.

New data types - interfaces mapping the SQL3 data types

Custom mapping of user-defined types (UTDs)

Miscellaneous features, including performance hints, the use of character streams, full precision for java.math.BigDecimal values, additional security, and support for time zones in date, time, and timestamp values.

Question165: Explain garbage collection?

Answer: Garbage collection is one of the most important feature of Java. Garbage collection is also called automatic memory management as JVM automatically removes the unused variables/objects (value is null) from the memory. User program cann't directly free the object from memory, instead it is the job of the garbage collector to automatically free the objects that are no longer referenced by a program. Every class inherits finalize() method from java.lang.Object, the finalize() method is called by garbage collector when it determines no more references to the object exists. In Java, it is good idea to explicitly assign null into a variable when no more in use. I Java on calling System.gc() and Runtime.gc(), JVM tries to recycle the unused objects, but there is no guarantee when all the objects will garbage collected.

Question166: How you can force the garbage collection?

Answer: Garbage collection automatic process and can't be forced.

Question167: What is OOPS?

Answer: OOP is the common abbreviation for Object-Oriented Programming.

Question168: Describe the principles of OOPS.

Answer: There are three main principals of oops which are called Polymorphism, Inheritance and Encapsulation.

Questio169: Explain the Encapsulation principle.

Answer: Encapsulation is a process of binding or wrapping the data and the codes that operates on the data into a single entity. This keeps the data safe from outside interface and misuse. One way to think about encapsulation is as a protective wrapper that prevents code and data from being arbitrarily accessed by other code defined outside the wrapper.

Question170: Explain the Inheritance principle.

Answer: Inheritance is the process by which one object acquires the properties of another object.

Question171: Explain the Polymorphism principle.

Answer: The meaning of Polymorphism is something like one name many forms. Polymorphism enables one entity to be used as as general category for different types of actions. The specific action is determined by the exact nature of the situation. The concept of polymorphism can be explained as "one interface, multiple methods".

Question172: Explain the different forms of Polymorphism.

Answer: From a practical programming viewpoint, polymorphism exists in three distinct forms in Java:

Method overloading

Method overriding through inheritance

Method overriding through the Java interface

Question173: What are Access Specifiers available in Java?

Answer: Access specifiers are keywords that determines the type of access to the member of a class. These are:

Public

Protected

Private

Defaults

Question174: what is the class variables ?

Answer: When we create a number of objects of the same class, then each object will share a common copy of variables. That means that there is only one copy per class, no matter how many objects are created from it. Class variables or static variables are declared with the static keyword in a class, but mind it that it should be declared outside outside a class. These variables are stored in static memory. Class variables are mostly used for constants, variable that never change its initial value. Static variables are always called by the class name. This variable is created when the program starts i.e. it is created before the instance is created of class by using new operator and gets destroyed when the programs stops. The scope of the class variable is same a instance variable. The class variable can be defined anywhere at class level with the keyword static. It initial value is same as instance variable. When the class variable is defined as int then it's initial value is by default zero, when declared boolean its default value is false and null for object references. Class variables are associated with the class, rather than with any object.

Question175: What is the difference between the instanceof and getclass, these two are same or not ?

Answer: instanceof is a operator, not a function while getClass is a method of java.lang.Object class. Consider a condition where we use

if(o.getClass().getName().equals("java.lang.Math")){ }

This method only checks if the classname we have passed is equal to java.lang.Math. The class java.lang.Math is loaded by the bootstrap ClassLoader. This class is an abstract class.This class loader is responsible for loading classes. Every Class object contains a reference to the ClassLoader that defines. getClass() method returns the runtime class of an object. It fetches the java instance of the given fully qualified type name. The code we have written is not necessary, because we should not compare getClass.getName(). The reason behind it is that if the two different class loaders load the same class but for the JVM, it will consider both classes as different classes so, we can't compare their names. It can only gives the implementing class but can't compare a interface, but instanceof operator can.

The instanceof operator compares an object to a specified type. We can use it to test if an object is an instance of a class, an instance of a subclass, or an instance of a class that implements a particular interface. We should try to use instanceof operator in place of getClass() method. Remember instanceof opeator and getClass are not same. Try this example, it will help you to better understand the difference between the two.

Interface one{

}

Class Two implements one {

}

Class Three implements one {

}

public class Test {

public static void main(String args[]) {

one test1 = new Two();

one test2 = new Three();

System.out.println(test1 instanceof one); //true

System.out.println(test2 instanceof one); //true

System.out.println(Test.getClass().equals(test2.getClass())); //false

}

}

\* Q176. Can an inner class declared inside of a method access local variables of this method?

A. It's possible if these variables are final.

\* Q177. What can go wrong if you replace && with & in the following code:

String a=null; if (a!=null && a.length()>10) {...}

A. A single ampersand here would lead to a NullPointerException.

\* Q178. What's the main difference between a Vector and an ArrayList

A. Java Vector class is internally synchronized and ArrayList is not.

\* Q179. When should the method invokeLater()be used?

A. This method is used to ensure that Swing components are updated through the event-dispatching thread.

\* Q180. How can a subclass call a method or a constructor defined in a superclass?

A. Use the following syntax: super.myMethod(); To call a constructor of the superclass, just write super(); in the first line of the subclass's constructor.

For senior-level developers:

\*\* Q181. What's the difference between a queue and a stack?

A. Stacks works by last-in-first-out rule (LIFO), while queues use the FIFO rule

\*\* Q182. You can create an abstract class that contains only abstract methods. On the other hand, you can create an interface that declares the same methods. So can you use abstract classes instead of interfaces?

A. Sometimes. But your class may be a descendent of another class and in this case the interface is your only option.

\*\* Q183. If you're overriding the method equals() of an object, which other method you might also consider?

A. hashCode()

\*\* Q184. You are planning to do an indexed search in a list of objects. Which of the two Java collections should you use:ArrayList or LinkedList?

A. ArrayList

\*\* Q185. How would you make a copy of an entire Java object with its state?

A. Have this class implement Cloneable interface and call its method clone().

\*\* Q186. How can you minimize the need of garbage collection and make the memory use more effective?

A. Use object pooling and weak object references.

\*\* Q187. There are two classes: A and B. The class B need to inform a class A when some important event has happened. What Java technique would you use to implement it?

A. If these classes are threads I'd consider notify() or notifyAll(). For regular classes you can use the Observer interface.

\*\* Q188. What access level do you need to specify in the class declaration to ensure that only classes from the same directory can access it?

A. You do not need to specify any access level, and Java will use a default package access level.

2. What are java beans?

JavaBeans is a portable, platform-independent component model written in the Java programming language, developed in collaboration with industry leaders. It enables developers to write reusable components once and run them anywhere -- benefiting from the platform-independent power of Java technology. JavaBeans acts as a Bridge between proprietary component models and provides a seamless and powerful means for developers to build components that run in ActiveX container applications.

Java beans is very powerful tool you can use in your servlet/JSP bridge. You can use the servlets to build the bean and can be passed over to the JSP for reading. This provides tight encapsulation of the data while preserving the sanctity of servlets and JSP.

3. What is RMI?

RMI stands for Remote Method Invocation. Traditional approaches to executing code on other machines across a network have been confusing as well as tedious and error-prone to implement. The nicest way to think about this problem is that some object happens to live on another machine, and that you can send a message to the remote object and get a result as if the object lived on your local machine. This simplification is exactly what Java Remote Method Invocation (RMI) allows you to do.

4. What gives java it's "write once and run anywhere" nature?

Java is compiled to be a byte code which is the intermediate language between source code and machine code. This byte code is not platorm specific and hence can be fed to any platform. After being fed to the JVM, which is specific to a particular operating system, the code platform. specific machine code is generated thus making java platform. independent.

5. How does Java inheritance work?

A class can only directly extend one class at a time. Multiple inheritance is only allowed with regard to interfaces. A class can implement many interfaces. But a class can only extend one non-interface class.

6. What are native methods? How do you use them?

Native methods are used when the implementation of a particular method is present in language other than Java say C, C++.

To use the native methods in java we use the keyword native

public native method\_a()

This native keyword is signal to the java compiler that the implementation of this method is in a language other than java.

Native methods are used when we realize that it would take up a lot of rework to write that piece of already existing code in other language to java.

7. Class A subclass B subclass C. All override foo(). I cast C to A and call foo(). What happens? Can C call A->foo()?

An instance of Class C is of type Class B and A (both). SO you can cast C to A. You CANNOT cast an instance of A to C.

8. What does the "static" keyword mean in front of a variable? A method? A class? Curly braces {}?

-- static variables: These are class level variable whose value remain same irrespective of the number of instances of the class.

-- static methods:

These are those methods that can be called without the need for creating the objects of the class i.e. they are class level methods. They can call only static methods. They cannot refer to "this" as they are not associated with any particular instance.

-- static block: These are called before the main is called and are called only once. Subsequent invocation of the java program containing static block would not call it again. Hence, they can be used to load libraries say in native function call.

-- Only Inner class could be declared as a "static". This declaration suppress the generation of the reference to the outer class object. 这意味着：1）为创建一个static内部类的对象，我们不需要一个外部类对象；2）不能从static内部类对象访问一个外部类对象。

9. How many different types of JDBC drivers are present? Discuss them.

There are four JDBC driver types.

Type 1: JDBC-ODBC Bridge plus ODBC Driver:

The first type of JDBC driver is the JDBC-ODBC Bridge. It is a driver that provides JDBC access to databases through ODBC drivers. The ODBC driver must be configured on the client for the bridge to work. This driver type is commonly used for prototyping or when there is no JDBC driver available for a particular DBMS.

Type 2: Native-API partly-Java Driver:

The Native to API driver converts JDBC commands to DBMS-specific native calls. This is much like the restriction of Type 1 drivers. The client must have some binary code loaded on its machine. These drivers do have an advantage over Type 1 drivers because they interface directly with the database.

Type 3: JDBC-Net Pure Java Driver:

The JDBC-Net drivers are a three-tier solution. This type of driver translates JDBC calls into a database-independent network protocol that is sent to a middleware server. This server then translates this DBMS-independent protocol into a DBMS-specific protocol, which is sent to a particular database. The results are then routed back through the middleware server and sent back to the client. This type of solution makes it possible to implement a pure Java client. It also makes it possible to swap databases without affecting the client.

Type 4: Native-Protocol Pure Java Driver

These are pure Java drivers that communicate directly with the vendor's database. They do this by converting JDBC commands directly into the database engine's native protocol. This driver has no additional translation or middleware layer, which improves performance tremendously.

10. Does Java have "goto"?

Yes and No. There is no "goto" operator used in Java, but it is a reserved keyword, and one can use break statements to branch to a labelled statement, exactly as one would use a goto.

11. Why "bytecode"? Can you reverse-engineer the code from bytecode?

yes, with some tools.

12. How does exception handling work in Java?

1.It separates the working/functional code from the error-handling code by way of try-catch clauses.

2.It allows a clean path for error propagation. If the called method encounters a situation it can't manage, it can throw an

exception and let the calling method deal with it.

3.By enlisting the compiler to ensure that "exceptional" situations are anticipated and accounted for, it enforces powerful coding.

4.Exceptions are of two types: Compiler-enforced exceptions, or checked exceptions and Runtime exceptions, or unchecked exceptions. Compiler-enforced (checked) exceptions are instances of the Exception class or one of its subclasses -- excluding the RuntimeException branch. The compiler expects all checked exceptions to be appropriately handled. Checked exceptions must be declared in the throws clause of the method throwing them -- assuming, of course, they're not being caught within that same method. The calling method must take care of these exceptions by either catching or declaring them in its throws clause. Thus, making an exception checked forces us to pay heed to the possibility of it being thrown. An example of a checked exception is java.io.IOException. As the name suggests, it throws whenever an input/output operation is abnormally terminated.

13. Does Java have destructors?

Java does not have destructors. Garbage collector does this job periodically depending upon the memory requirements of the machine and on the fact that a particular object is no longer needed.

But it has finalizers that does a similar job. The syntax is

public void finalize() { }

If an object has a finalizer, the method is invoked before the system garbage collects the object, but using finalize() does not guarantee that it would be called b4 garbage collector is invoked.

14. What does the "final" keyword mean in front of a variable? A method? A class?

A final variable cannot be reassigned, but it is not constant. For instance,

final StringBuffer x = new StringBuffer();

x.append("hello");

is valid. X cannot have a new value in it, but nothing stops operations on the object that it refers, including destructive operations.

Also, a final method cannot be overridden or hidden by new access specifications. This means that the compiler can choose to in-line the invocation of such a method. (I don't know if any compiler actually does this, but it's true in theory.)

The best example of a final class is String, which defines a class that cannot be derived.

15. Access specifiers: "public", "protected", "private", nothing?

Public? Any other class from any package can instantiate and execute the classes and methods

Protected? Only subclasses and classes inside of the package can access the classes and methods

Private? The original class is the only class allowed to execute the methods.

And in case if there is no modifier specified, it means, only the classes inside the package can access this class and its methods, it is also called "Friendly".

2. What are java beans?

JavaBeans is a portable, platform-independent component model written in the Java programming language, developed in collaboration with industry leaders. It enables developers to write reusable components once and run them anywhere -- benefiting from the platform-independent power of Java technology. JavaBeans acts as a Bridge between proprietary component models and provides a seamless and powerful means for developers to build components that run in ActiveX container applications.

Java beans is very powerful tool you can use in your servlet/JSP bridge. You can use the servlets to build the bean and can be passed over to the JSP for reading. This provides tight encapsulation of the data while preserving the sanctity of servlets and JSP.

3. What is RMI?

RMI stands for Remote Method Invocation. Traditional approaches to executing code on other machines across a network have been confusing as well as tedious and error-prone to implement. The nicest way to think about this problem is that some object happens to live on another machine, and that you can send a message to the remote object and get a result as if the object lived on your local machine. This simplification is exactly what Java Remote Method Invocation (RMI) allows you to do.

4. What gives java it's "write once and run anywhere" nature?

Java is compiled to be a byte code which is the intermediate language between source code and machine code. This byte code is not platorm specific and hence can be fed to any platform. After being fed to the JVM, which is specific to a particular operating system, the code platform. specific machine code is generated thus making java platform. independent.

5. How does Java inheritance work?

A class can only directly extend one class at a time. Multiple inheritance is only allowed with regard to interfaces. A class can implement many interfaces. But a class can only extend one non-interface class.

6. What are native methods? How do you use them?

Native methods are used when the implementation of a particular method is present in language other than Java say C, C++.

To use the native methods in java we use the keyword native

public native method\_a()

This native keyword is signal to the java compiler that the implementation of this method is in a language other than java.

Native methods are used when we realize that it would take up a lot of rework to write that piece of already existing code in other language to java.

7. Class A subclass B subclass C. All override foo(). I cast C to A and call foo(). What happens? Can C call A->foo()?

An instance of Class C is of type Class B and A (both). SO you can cast C to A. You CANNOT cast an instance of A to C.

8. What does the "static" keyword mean in front of a variable? A method? A class? Curly braces {}?

-- static variables: These are class level variable whose value remain same irrespective of the number of instances of the class.

-- static methods:

These are those methods that can be called without the need for creating the objects of the class i.e. they are class level methods. They can call only static methods. They cannot refer to "this" as they are not associated with any particular instance.

-- static block: These are called before the main is called and are called only once. Subsequent invocation of the java program containing static block would not call it again. Hence, they can be used to load libraries say in native function call.

-- Only Inner class could be declared as a "static". This declaration suppress the generation of the reference to the outer class object. 这意味着：1）为创建一个static内部类的对象，我们不需要一个外部类对象；2）不能从static内部类对象访问一个外部类对象。

9. How many different types of JDBC drivers are present? Discuss them.

There are four JDBC driver types.

Type 1: JDBC-ODBC Bridge plus ODBC Driver:

The first type of JDBC driver is the JDBC-ODBC Bridge. It is a driver that provides JDBC access to databases through ODBC drivers. The ODBC driver must be configured on the client for the bridge to work. This driver type is commonly used for prototyping or when there is no JDBC driver available for a particular DBMS.

Type 2: Native-API partly-Java Driver:

The Native to API driver converts JDBC commands to DBMS-specific native calls. This is much like the restriction of Type 1 drivers. The client must have some binary code loaded on its machine. These drivers do have an advantage over Type 1 drivers because they interface directly with the database.

Type 3: JDBC-Net Pure Java Driver:

The JDBC-Net drivers are a three-tier solution. This type of driver translates JDBC calls into a database-independent network protocol that is sent to a middleware server. This server then translates this DBMS-independent protocol into a DBMS-specific protocol, which is sent to a particular database. The results are then routed back through the middleware server and sent back to the client. This type of solution makes it possible to implement a pure Java client. It also makes it possible to swap databases without affecting the client.

Type 4: Native-Protocol Pure Java Driver

These are pure Java drivers that communicate directly with the vendor's database. They do this by converting JDBC commands directly into the database engine's native protocol. This driver has no additional translation or middleware layer, which improves performance tremendously.

10. Does Java have "goto"?

Yes and No. There is no "goto" operator used in Java, but it is a reserved keyword, and one can use break statements to branch to a labelled statement, exactly as one would use a goto.

11. Why "bytecode"? Can you reverse-engineer the code from bytecode?

yes, with some tools.

12. How does exception handling work in Java?

1.It separates the working/functional code from the error-handling code by way of try-catch clauses.

2.It allows a clean path for error propagation. If the called method encounters a situation it can't manage, it can throw an

exception and let the calling method deal with it.

3.By enlisting the compiler to ensure that "exceptional" situations are anticipated and accounted for, it enforces powerful coding.

4.Exceptions are of two types: Compiler-enforced exceptions, or checked exceptions and Runtime exceptions, or unchecked exceptions. Compiler-enforced (checked) exceptions are instances of the Exception class or one of its subclasses -- excluding the RuntimeException branch. The compiler expects all checked exceptions to be appropriately handled. Checked exceptions must be declared in the throws clause of the method throwing them -- assuming, of course, they're not being caught within that same method. The calling method must take care of these exceptions by either catching or declaring them in its throws clause. Thus, making an exception checked forces us to pay heed to the possibility of it being thrown. An example of a checked exception is java.io.IOException. As the name suggests, it throws whenever an input/output operation is abnormally terminated.

13. Does Java have destructors?

Java does not have destructors. Garbage collector does this job periodically depending upon the memory requirements of the machine and on the fact that a particular object is no longer needed.

But it has finalizers that does a similar job. The syntax is

public void finalize() { }

If an object has a finalizer, the method is invoked before the system garbage collects the object, but using finalize() does not guarantee that it would be called b4 garbage collector is invoked.

14. What does the "final" keyword mean in front of a variable? A method? A class?

A final variable cannot be reassigned, but it is not constant. For instance,

final StringBuffer x = new StringBuffer();

x.append("hello");

is valid. X cannot have a new value in it, but nothing stops operations on the object that it refers, including destructive operations.

Also, a final method cannot be overridden or hidden by new access specifications. This means that the compiler can choose to in-line the invocation of such a method. (I don't know if any compiler actually does this, but it's true in theory.)

The best example of a final class is String, which defines a class that cannot be derived.

15. Access specifiers: "public", "protected", "private", nothing?

Public? Any other class from any package can instantiate and execute the classes and methods

Protected? Only subclasses and classes inside of the package can access the classes and methods

Private? The original class is the only class allowed to execute the methods.

And in case if there is no modifier specified, it means, only the classes inside the package can access this class and its methods, it is also called "Friendly".