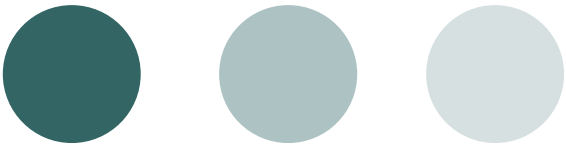
First edition (early access)



Frequently Asked Technical Interview Questions and Answers Guide

www.acetheinterview.com

TABLE OF CONTENT

Part-I: Frequently Asked Interview Questions

Questions in an Information Technology related job interview can be organized into the following categories:

1. General
2. Fundamental
3. Analytical
4. Algorithms & Coding
5. Language/Technology specifics

The questions are more or less designed around the specific position you are interested in; however, you should expect some common questions irrespective of the job description. As in any typical interview, the questions below are not arranged in any particular order of difficulty. Please try to answer these questions on your own before looking at the sample answers in Part-II, as it will go a long way in getting you ready for your interview. In addition, keep in mind that this is not meant to be an exhaustive list of Interview questions- just some of the most frequently asked questions. You are encouraged to consult other sources as part of your interview preparation.

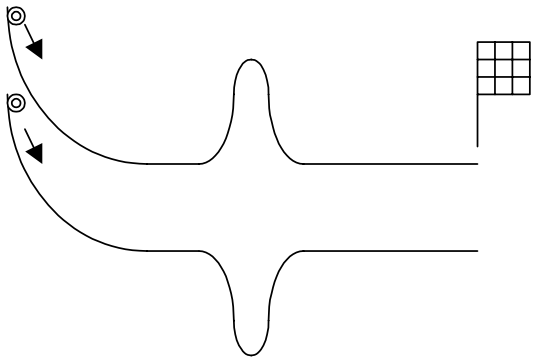
14) There was a sheriff in a town that caught three outlaws. He said he was going to give them all a chance to go free. All they had to do is figure out what color hat they were wearing. The sheriff had 5 hats, 3 black and 2 white. Each outlaw can see the color of the other outlaw's hats, but cannot see his own. The first outlaw guessed and was wrong so he was put in jail. The second outlaw also guessed and was also put in jail. Finally the third blind outlaw guessed and he guessed correctly. How did he know? (Answer)

15) One train leaves Los Angeles at 15 MPH heading for New York. Another train leaves from New York at 20mph heading for Los Angeles on the same track. If a bird, flying at 25mph, leaves from Los Angeles at the same time as the train and flies back and forth between the two trains until they collide, how far will the bird have traveled? (Answer)

16) Imagine that you have 26 constants, labeled A through Z. Each constant is assigned a value in the following way: A = 1; the rest of the values equal their position in the alphabet (B corresponds to the second position so it equals 2, C = 3, etc.) raised to the power of the preceding constant value. So, B = 2 ^ (A's value), or B = 2^1 = 2. C = 3^2 = 9. D = 4^9, etc., etc. Find the exact numerical value to the following equation:

(X - A) * (X - B) * (X - C) * ... * (X - Y) * (X - Z) (Answer)

17) Two balls begin rolling down the tracks below (see diagram). Which one reaches the end first, and which is moving faster at the end? Assume no friction or wind resistance. (Answer)



18) A farmer has $100.00. He wants 100 animals. What combination of the animals below will get the farmer EXACTLY 100 animals with EXACTLY $100.00? Cows are $2.00 each, Pigs are $0.50 each and Chickens are $0.10 each.

4. ALGORITHMS & CODING

1) You are developing a web browser (something like e.g. Netscape, etc.) and need to display all visited links on a page. The visited links need to use a different color then that used to display scheme than the unvisited links. Now, given a history of links you have visited before, how would you go about writing the piece of code that makes the determination if you have seen this link before? Answer or not? The answer could be a simple string comparison, but then think about the time it will take for the client to render any HTML page. Alternatively, so, given a history of URLs, come-up with an elegant way (algorithm, data structures, etc.) to make the determination if a given link already exists in the history list? (Answer)
2) Since web pages can have multiple URLs pointing to them, as a web browser developer how can you make sure you have never seen the same content before? (Answer)
3) Write a function to print all the possible permutations of a string. Now, modify the algorithm to discard duplicates. (Answer)
4) Design a memory management scheme.
5) Implement strstr(), strcpy(), strtok() etc. (Answer)
6) Write an algorithm and C code to find the sub array with the largest sum given an array that contains both positive and negative integers (Answer)
7) A square picture is cut into 16 squares and then shuffled. Write a program to rearrange the 16 squares to get the original big square. (Answer)
8) Implement an algorithm to reverse a singly linked list (with and without recursion). (Answer)
9) Count the total number of set bits in a number without using a loop. (Answer)
10) Given an array of characters which form a sentence of words, give an efficient algorithm to reverse the order of the words in it. (Answer)
11) Do the class/structure description for a Hash table, and write the source code for the insert function. (Answer)
12) What sort of technique you would use to update a set of files over a network, where a server contains the master copy.
13) How do you handle deadlock on a table that is fed with a live serial feed?

15) The distance traveled by the bird can be calculated as:

1. Let's say the distance between LA and NY is d miles.
2. The time before which the trains will collide: d / (15+20) hours.
3. The distance traveled by the bird in that time: (d / 35) * 25 = 5*d/7 miles.

Assumptions made: The bird must follow the line of the track, remain at the same altitude, and the speed must be relative to the ground and not air speed.

(Return to question)

16) (X - A) * (X - B) * (X - C) * ... * (X - Y) * (X - Z) equals 0. This is because (X - X) is zero, and any integer multiplied by zero is zero. (Return to question)

17) The ball on the lower track gets there first. Both balls are traveling at the same speed at the end of the track. Hint: The ball on the lower track travels the same distance but at a higher average speed). (Return to question)

5) To see samples of this code you can look at the C Runtime code that ships along with VC++. The code is very efficient and well written. (Return to question)

6) The information to be kept while going through the array only once (O(n)) is the best sub array ending in the current position and best sub array found for the entire array until the current step. If necessary, the starting and ending position for the sub array can be kept. After the processing of the last element, the algorithm will have determined the sub array having the largest sum. The algorithm works fine even if there is no negative number in the array.

Algorithm: "arr" is the array of n integer. The function will return the largest sum and the limits of the sub array producing that value.

```
int GetBestSubArray(int* arr , int n , int* nBegin , int* nEnd)
{
   //starting/ending position of the best sub array for entire array
   int nBestBegin=0,nBestEnd=0;

   //starting/ending position of the best sub array ending in
   //current position

   int nCrtBegin=0,nCrtEnd=0;
   // index to loop in the array

   int nCrtIndex = 0;
   //the sum of whole best sub array and the sum of current best
   //sub array
   int nBestSum=0, nCrtSum=0;
   //Nothing to analyze, return invalid array indexes
   if (n == 0){
        (*nEnd)=(*nBegin)=-1;
        return 0;
   }
   nBestSum=nCrtSum = arr[nCrtIndex];

   for(nCrtIndex=1;nCrtIndex<n;nCrtIndex++) {
        //Compute the current largest sum
        if (nCrtSum<0) {
          nCrtSum = arr[nCrtIndex];
          nCrtEnd = nCrtBegin=nCrtIndex;
```



```
}
```

You can also use a lookup table to count the number of set bits.

(Return to question)

10) One way to reverse the order of words is:

```
string word_reverse(char * data)
{
   char c = ' ';
   string s=data;
   int len=strlen(data);
   vector<string> output;
   int i=0;
   while(s[i]!='\0') {
      if(s[i]==c) {
         output.push_back(s.substr(0,i));
         s=s.substr(i+1,len);
         i=0;
         continue;
      }
      i++;
   }
   output.push_back(s);
   s.erase(0,s.length());
   string s1;
   for(int j=output.size()-1;j>0;j--) {
      s1+=output[j];
      s1+=" ";
   }
   s1+=output[0];
   return s1;
}
```

(Return to question)

11) A very simplistic and limited implementation of Hash table may look something like this:

For a 100 entry hash table of words, put in the arguments from the command line into the hash table:

```
#define MAX 100
```


pointers." You can make up your mind whether it's really a pointer or not. In any event, there's no pointer arithmetic.

- There are no destructors in Java (automatic garbage collection).
- Java does not support conditional compile (#ifdef/#ifndef type).
- Thread support is built into java but not in C++.
- Java does not support default arguments.
- There's no scope resolution operator: in Java. Java uses the dot for everything, but can get away with it since you can define elements only within a class. Even the method definitions must always occur within a class, so there is no need for scope resolution there either.
- There's no "goto" statement in Java.
- Java doesn't provide multiple inheritance (MI), at least not in the same sense that C++ does.
- Exception handling in Java is different because there are no destructors.
- Java has method overloading, but no operator overloading. The String class does use the + and += operators to concatenate strings and String expressions use automatic type conversion, but that's a special built-in case.
- Java is interpreted for the most part and hence platform independent.

(Return to question)

5) Interfaces provide more sophisticated ways to organize and control the objects in your system. The interface keyword takes the abstract concept one step further. You could think of it as a "pure" abstract class. It allows the creator to establish the form for a class: method names, argument lists, and return types, but no method bodies. An interface can also contain fields. An interface says: "This is what all classes that implement this particular interface will look like." Thus, any code that uses a particular interface knows what methods might be called for that interface, and that's all. So the interface is used to establish a "protocol" between classes (some object-oriented programming languages have a keyword called protocol to do the same thing).

A typical example is listed below (from "Thinking in Java" by Bruce Eckels):

7) Also called a "proxy" or "application level gateway," it is an application that breaks the connection between sender and receiver. All input is forwarded out a different port, closing a straight path between two networks and preventing a hacker from obtaining internal addresses and details of a private network. Proxy servers are available for common Internet services; for example, an HTTP proxy is used for Web access, and an SMTP proxy is used for e-mail. Proxies generally employ network address translation (NAT), which presents one organization-wide IP address to the Internet. It funnels all user requests to the Internet and fans responses back out to the appropriate users. Proxies may also cache Web pages, so that the next request can be obtained locally. Proxies are only one tool that can be used to build a firewall. (Return to question)

8) Cookies are small pieces of data used by web servers to help identify web users. There two types of cookies: Persistent and Non-Persistent.

- Persistent: Which remains on the user's machine even after user leaves the site. When they come back web server queries the cookie and if found identifies the user
- Non-Persistent: This cookie is used by web server to track the user information during a session. It expires once you leave the website

(Return to question)

9) The URL of the link is given to the browser engine, which opens a TCP/IP connection to the address listed in the URL. Assuming the URL specifies the HTTP protocol, and assuming no port is given in the URL, the connection is made to port 80 of the remote host. Various options may be specified in the HTTP headers which effect the expected behavior between client and host, an important and common option being "Connection: Keep-alive"; if this option is specified, the same connection is used to transfer all resources on the page (images, frames, sounds, flash animations, etc.), thus avoiding the overhead of establishing a separate TCP/IP connection for each resource the page requires to be loaded. Requests are made by issuing an HTTP "GET" command, followed by the resource requested. A single GET command may cause the client to request multiple resources (hence the existence of the keep-alive option). Finally, under normal circumstances, the connection

