IT Interview Questions

- Contents –

Technical

Non-Technical

C Interview Questions

1. What is C language?

The C programming language is a standardized programming language developed in the early 1970s by Ken Thompson and Dennis Ritchie for use on the UNIX operating system. It has since spread to many other operating systems, and is one of the most widely used programming languages. C is prized for its efficiency, and is the most popular programming language for writing system software, though it is also used for writing applications.

2. What does static variable mean?

There are 3 main uses for the static.

1. If you declare within a function: It retains the value between function calls

2. If it is declared for a function name: By default function is extern..so it will be visible from other files if the function declaration is as static..it is invisible for the outer files

3. Static for global variables: By default we can use the global variables from outside files If it is static global..that variable is limited to with in the file.

```
#include <stdio.h>
int t = 10;
main(){
int x = 0;
void funct1();
funct1();
printf("After first call \n");
funct1();
printf("After second call \n");
funct1();
printf("After third call \n");
}
void funct1()
{
   static int y = 0;
   int z = 10;
   printf("value of y %d z %d",y,z);
   y=y+10;
}
```

value of y 0 z 10 After first call
value of y 10 z 10 After second call
value of y 20 z 10 After third call

3. What are the different storage classes in C?

C has three types of storage: automatic, static and allocated. Variable having block scope and without static specifier have automatic storage duration.

Variables with block scope, and with static specifier have static scope. Global variables (i.e, file scope) with or without the the static specifier also have static scope. Memory obtained from calls to malloc(), alloc() or realloc() belongs to allocated storage class.

4. What is hashing?

To hash means to grind up, and that's essentially what hashing is all about. The heart of a hashing algorithm is a hash function that takes your nice, neat data and grinds it into some random-looking integer.

The idea behind hashing is that some data either has no inherent ordering (such as images) or is expensive to compare (such as images). If the data has no inherent ordering, you can't perform comparison searches.

5. Can static variables be declared in a header file?

You can't declare a static variable without defining it as well (this is because the storage class modifiers static and extern are mutually exclusive). A static variable can be defined in a header file, but this would cause each source file that included the header file to have its own private copy of the variable, which is probably not what was intended.

6. Can a variable be both constant and volatile?

Yes. The const modifier means that this code cannot change the value of the variable, but that does not mean that the value cannot be changed by means outside this code.

The function itself did not change the value of the timer, so it was declared const. However, the value was changed by hardware on the computer, so it was declared volatile. If a variable is both const and volatile, the two modifiers can appear in either order.

7. Can include files be nested?

Yes. Include files can be nested any number of times. As long as you use precautionary measures, you can avoid including the same file twice. In the past, nesting header files was seen as bad programming practice, because it complicates the dependency tracking function of the MAKE program and thus slows down compilation. Many of today's popular compilers make up for this difficulty by implementing a concept called precompiled headers, in which all headers and associated dependencies are stored in a precompiled state.

8. What is a null pointer?

There are times when it's necessary to have a pointer that doesn't point to anything. The macro NULL, defined in , has a value that's guaranteed to be different from any valid pointer. NULL is a literal zero, possibly cast to void* or char*.

Some people, notably C++ programmers, prefer to use 0 rather than NULL.
The null pointer is used in three ways:
1) To stop indirection in a recursive data structure.
2) As an error value.
3) As a sentinel value.

9. What is the output of printf("%d") ?

When we write printf("%d",x); this means compiler will print the value of x. But as here, there is nothing after %d so compiler will show in output window garbage value.

10. What is the difference between calloc() and malloc() ?

calloc(...) allocates a block of memory for an array of elements of a certain size. By default the block is initialized to 0. The total number of memory allocated will be (number_of_elements * size).

malloc(...) takes in only a single argument which is the memory required in bytes. malloc(...) allocated bytes of memory and not blocks of memory like calloc(...).

malloc(...) allocates memory blocks and returns a void pointer to the allocated space, or NULL if there is insufficient memory available.

calloc(...) allocates an array in memory with elements initialized to 0 and returns a pointer to the allocated space. calloc(...) calls malloc(...) in order to use the C++ _set_new_mode function to set the new handler mode.

11. What is the difference between printf() and sprintf() ?

sprintf() writes data to the character array whereas printf(...) writes data to the standard output device.

12. How to reduce a final size of executable?

Size of the final executable can be reduced using dynamic linking for libraries.

13. Can you tell me how to check whether a linked list is circular?

Create two pointers, and set both to the start of the list. Update each as follows:

```
while (pointer1) {
pointer1 = pointer1->next;
pointer2 = pointer2->next;
if (pointer2) pointer2=pointer2->next;
if (pointer1 == pointer2) {
print ("circular");
}
}
```

If a list is circular, at some point pointer2 will wrap around and be either at the item just before pointer1, or the item before that. Either way, its either 1 or 2 jumps until they meet.

14. Advantages of a macro over a function?

Macro gets to see the Compilation environment, so it can expand __ __TIME__ __FILE__ #defines. It is expanded by the preprocessor.

For example, you can't do this without macros
#define PRINT(EXPR) printf(#EXPR "=%d\n", EXPR)
PRINT(5+6*7) // expands into printf("5+6*7=%d", 5+6*7);
You can define your mini language with macros:
#define strequal(A,B) (!strcmp(A,B))

15. What is the difference between strings and character arrays?

A major difference is: string will have static storage duration, whereas as a character array will not, unless it is explicity specified by using the static keyword.

Actually, a string is a character array with following properties:
* the multibyte character sequence, to which we generally call string, is used to initialize an array of static storage duration. The size of this array is just sufficient to contain these

characters plus the terminating NUL character.
* it not specified what happens if this array, i.e., string, is modified.
* Two strings of same value[1] may share same memory area.

16. Write down the equivalent pointer expression for referring the same element a[i][j][k][l] ?

a[i] == *(a+i)
a[i][j] == *(*(a+i)+j)
a[i][j][k] == *(*(*(a+i)+j)+k)
a[i][j][k][l] == *(*(*(*(a+i)+j)+k)+l)

17. Which bit wise operator is suitable for checking whether a particular bit is on or off?

The bitwise AND operator. Here is an example:

```
enum {
KBit0 = 1,
KBit1,
…
KBit31,
};
if ( some_int & KBit24 )
printf ( “Bit number 24 is ON\n” );
else
printf ( “Bit number 24 is OFF\n” );
```

18. Which bit wise operator is suitable for turning off a particular bit in a number?

The bitwise AND operator, again. In the following code snippet, the bit number 24 is reset to zero.

```
some_int = some_int & ~KBit24;
```

19. Which bit wise operator is suitable for putting on a particular bit in a number?

The bitwise OR operator. In the following code snippet, the bit number 24 is turned ON:
some_int = some_int | KBit24;

20. Does there exist any other function which can be used to convert an integer or a float to a string?

Some implementations provide a nonstandard function called itoa(), which converts an integer to string.

#include
char *itoa(int value, char *string, int radix);
DESCRIPTION
The itoa() function constructs a string representation of an integer.
PARAMETERS
value: Is the integer to be converted to string representation.
string: Points to the buffer that is to hold resulting string.
The resulting string may be as long as seventeen bytes.
radix: Is the base of the number; must be in the range 2 - 36.
A portable solution exists. One can use sprintf():
char s[SOME_CONST];
int i = 10;
float f = 10.20;
sprintf (s, "%d %f\n", i, f);

21. Why does malloc(0) return valid memory address ? What's the use?

malloc(0) does not return a non-NULL under every implementation. An implementation is free to behave in a manner it finds suitable, if the allocation size requested is zero.
The implmentation may choose any of the following actions:

* A null pointer is returned.
* The behavior is same as if a space of non-zero size was requested. In this case, the usage of return value yields to undefined-behavior.

Notice, however, that if the implementation returns a non-NULL value for a request of a zero-length space, a pointer to object of ZERO length is returned! Think, how an object of zero size should be represented

For implementations that return non-NULL values, a typical usage is as follows:
void
func (void)
{
int *p; /* p is a one-dimensional array, whose size will vary during the the lifetime of the program */
size_t c;
p = malloc(0); /* initial allocation */
if (!p)
{
perror ("FAILURE");
return;
}

```
/* … */
while (1)
{
c = (size_t) … ; /* Calculate allocation size */
p = realloc ( p, c * sizeof *p );
/* use p, or break from the loop */
/* … */
}
return;
}
```

Notice that this program is not portable, since an implementation is free to return NULL for a malloc(0) request, as the C Standard does not support zero-sized objects.

22. Difference between const char* p and char const* p

In const char* p, the character pointed by 'p' is constant, so u cant change the value of character pointed by p but u can make 'p' refer to some other location.

In char const* p, the ptr 'p' is constant not the character referenced by it, so u cant make 'p' to reference to any other location but u can change the value of the char pointed by 'p'.

23. What is the result of using Option Explicit?

When writing your C program, you can include files in two ways. The first way is to surround the file you want to include with the angled brackets < and >. This method of inclusion tells the preprocessor to look for the file in the predefined default location. This predefined default location is often an INCLUDE environment variable that denotes the path to your include files.

For instance, given the INCLUDE variable INCLUDE=C:\COMPILER\INCLUDE;S:\SOURCE\HEADERS; using the #include version of file inclusion, the compiler first checks the C:\COMPILER\INCLUDE directory for the specified file. If the file is not found there, the compiler then checks the S:\SOURCE\HEADERS directory. If the file is still not found, the preprocessor checks the current directory.

The second way to include files is to surround the file you want to include with double quotation marks. This method of inclusion tells the preprocessor to look for the file in the current directory first, then look for it in the predefined locations you have set up. Using the #include file version of file inclusion and applying it to the preceding example, the preprocessor first checks the current directory for the specified file. If the file is not found in the current directory, the C:COMPILERINCLUDE directory is searched. If the file is still not found, the preprocessor checks the S:SOURCEHEADERS directory.

The #include method of file inclusion is often used to include standard headers such as stdio.h or stdlib.h.

The #include file include nonstandard header files that you have created for use in your program. This is because these headers are often modified in the current directory, and you will want the preprocessor to use your newly modified version of the header rather than the older, unmodified version.

24. What is the benefit of using an enum rather than a #define constant?

The use of an enumeration constant (enum) has many advantages over using the traditional symbolic constant style of #define. These advantages include a lower maintenance requirement, improved program readability, and better debugging capability.
1) The first advantage is that enumerated constants are generated automatically by the compiler. Conversely, symbolic constants must be manually assigned values by the programmer.
2) Another advantage of using the enumeration constant method is that your programs are more readable and thus can be understood better by others who might have to update your program later.

3) A third advantage to using enumeration constants is that some symbolic debuggers can print the value of an enumeration constant. Conversely, most symbolic debuggers cannot print the value of a symbolic constant. This can be an enormous help in debugging your program, because if your program is stopped at a line that uses an enum, you can simply inspect that constant and instantly know its value. On the other hand, because most debuggers cannot print #define values, you would most likely have to search for that value by manually looking it up in a header file.

25. What is the quickest sorting method to use?

The answer depends on what you mean by quickest. For most sorting problems, it just doesn't matter how quick the sort is because it is done infrequently or other operations take significantly more time anyway. There are three sorting methods in this author's toolbox that are all very fast and that are useful in different situations. Those methods are quick sort, merge sort, and radix sort.

26. When should the volatile modifier be used?

The volatile modifier is a directive to the compiler's optimizer that operations involving this variable should not be optimized in certain ways. There are two special cases in which use of the volatile modifier is desirable. The first case involves memory-mapped hardware (a device such as a graphics adaptor that appears to the computer's hardware as if it were part of the computer's memory), and the second involves shared memory (memory used by two or more programs running simultaneously).

27. When should the register modifier be used?

The register modifier hints to the compiler that the variable will be heavily used and should be kept in the CPU's registers, if possible, so that it can be accessed faster.

28. How can you determine the size of an allocated portion of memory?

You can't, really. free() can , but there's no way for your program to know the trick free() uses. Even if you disassemble the library and discover the trick, there's no guarantee the trick won't change with the next release of the compiler.

29. When does the compiler not implicitly generate the address of the first element of an array?

Whenever an array name appears in an expression such as
· array as an operand of the size of operator
· array as an operand of & operator
· array as a string literal initializer for a character array
Then the compiler does not implicitly generate the address of the address of the first element of an array.

30. Why n++ executes faster than n+1 ?

The expression n++ requires a single machine instruction such as INR to carry out the increment operation whereas, n+1 requires more instructions to carry out this operation.

31. Why doesn't the following statement work?

```
char str[ ] = "Hello" ;
strcat ( str, '!' ) ;
```

Answer: The string function strcat( ) concatenates strings and not a character. The basic difference between a string and a character is that a string is a collection of characters, represented by an array of characters whereas a character is a single character. To make the above statement work writes the statement as shown below:

```
strcat ( str, "!" ) ;
```

32. What is the benefit of using #define to declare a constant?

Using the #define method of declaring a constant enables you to declare a constant in one place and use it throughout your program. This helps make your programs more maintainable, because you need to maintain only the #define statement and not several instances of individual constants throughout your program.

For instance, if your program used the value of pi (approximately 3.14159) several times, you might want to declare a constant for pi as follows: #define PI 3.14159

Using the #define method of declaring a constant is probably the most familiar way of declaring constants to traditional C programmers. Besides being the most common method of declaring constants, it also takes up the least memory.

Constants defined in this manner are simply placed directly into your source code, with no variable space allocated in memory. Unfortunately, this is one reason why most debuggers cannot inspect constants created using the #define method.

33. What is the purpose of main( ) function ?

The function main( ) invokes other functions within it.It is the first function to be called when the program starts execution.

· It is the starting function
· It returns an int value to the environment that called the program
· Recursive call is allowed for main( ) also.
· It is a user-defined function
· Program execution ends when the closing brace of the function main( ) is reached.
· It has two arguments 1)argument count and 2) argument vector (represents strings passed).
· Any user-defined name can also be used as parameters for main( ) instead of argc and argv

34. How can I search for data in a linked list?

Unfortunately, the only way to search a linked list is with a linear search, because the only way a linked list's members can be accessed is sequentially.
Sometimes it is quicker to take the data from a linked list and store it in a different data structure so that searches can be more efficient.

35. Why should we assign NULL to the elements (pointer) after freeing them?

This is paranoia based on long experience. After a pointer has been freed, you can no longer use the pointed-to data. The pointer is said to dangle; it doesn't point at anything useful.

If you NULL out or zero out a pointer immediately after freeing it, your program can no longer get in trouble by using that pointer. True, you might go indirect on the null pointer instead, but that's something your debugger might be able to help you with immediately.

Also, there still might be copies of the pointer that refer to the memory that has been deallocated; that's the nature of C. Zeroing out pointers after freeing them won't solve all problems.

36. What is a null pointer assignment error? What are bus errors, memory faults, and core dumps?

These are all serious errors, symptoms of a wild pointer or subscript. Null pointer assignment is a message you might get when an MS-DOS program finishes executing. Some such programs can arrange for a small amount of memory to be available "where the NULL pointer points to (so to speak). If the program tries to write to that area, it will overwrite the data put there by the compiler.

When the program is done, code generated by the compiler examines that area. If that data has been changed, the compiler-generated code complains with null pointer assignment. This message carries only enough information to get you worried. There's no way to tell, just from a null pointer assignment message, what part of your program is responsible for the error. Some debuggers, and some compilers, can give you more help in finding the problem.

Bus error: core dumped and Memory fault: core dumped are messages you might see from a program running under UNIX. They're more programmer friendly. Both mean that a pointer or an array subscript was wildly out of bounds. You can get these messages on a read or on a write. They aren't restricted to null pointer problems. The core dumped part of the message is telling you about a file, called core, that has just been written in your current directory. This is a dump of everything on the stack and in the heap at the time the program was running. With the help of a debugger, you can use the core dump to find where the bad pointer was used. That might not tell you why the pointer was bad, but it's a step in the right direction. If you don't have write permission in the current directory, you won't get a core file, or the core dumped message

37. Predict the output or error(s) for the following programmes:

```
void main()
{
int const * p=5;
printf("%d",++(*p));
}
```

Answer: Compiler error: Cannot modify a constant value.
Explanation: p is a pointer to a "constant integer". But we tried to change the value of the "constant integer".

```
38. main()
{
char s[ ]="man";
int i;
```

```
for(i=0;s[ i ];i++)
printf("\n%c%c%c%c",s[ i ],*(s+i),*(i+s),i[s]);
}
```

Answer:
mmm
aaaa
nnnn

Explanation: s[i], *(i+s), *(s+i), i[s] are all different ways of expressing the same idea. Generally array name is the base address for that array. Here s is the base address. i is the index number/ displacement from the base address. So, indirecting it with * is same as s[i]. i[s] may be surprising. But in the case of C it is same as s[i].

39. main()

```
{
float me = 1.1;
double you = 1.1;
if(me==you)
printf("I love U");
else
printf("I hate U");
}
```

Answer: I hate U
Explanation: For floating point numbers (float, double, long double) the values cannot be predicted exactly. Depending on the number of bytes, the precession with of the value represented varies. Float takes 4 bytes and long double takes 10 bytes. So float stores 0.9 with less precision than long double.

Rule of Thumb: Never compare or at-least be cautious when using floating point numbers with relational operators (== , >, <, <=, >=,!=) .

40. main()

```
{
static int var = 5;
printf("%d ",var--);

if(var)
main();
}
```

Answer: 5 4 3 2 1
Explanation: When static storage class is given, it is initialized once. The change in the value of a static variable is retained even between the function calls. Main is also treated like any other ordinary function, which can be called recursively.

41. main()

```
{
int c[ ]={2.8,3.4,4,6.7,5};
int j,*p=c,*q=c;
for(j=0;j<5;j++) {
printf(" %d ",*c);
++q; }
for(j=0;j<5;j++){
printf(" %d ",*p);
++p; }
}
```

Answer: 2 2 2 2 2 2 3 4 6 5

Explanation: Initially pointer c is assigned to both p and q. In the first loop, since only q is incremented and not c , the value 2 will be printed 5 times. In second loop p itself is incremented. So the values 2 3 4 6 5 will be printed.

42. main()

```
{
extern int i;
i=20;
printf("%d",i);
}
```

Answer: Linker Error : Undefined symbol '_i'

Explanation: extern storage class in the following declaration,
extern int i;
specifies to the compiler that the memory for i is allocated in some other program and that address will be given to the current program at the time of linking. But linker finds that no other variable of name i is available in any other program with memory space allocated for it. Hence a linker error has occurred .

43. main()

```
{
int i=-1,j=-1,k=0,l=2,m;
m=i++&&j++&&k++||l++;
printf("%d %d %d %d %d",i,j,k,l,m);
}
```

Answer: 0 0 1 3 1

Explanation: Logical operations always give a result of 1 or 0. And also the logical AND (&&) operator has higher priority over the logical OR (||) operator. So the expression 'i++ && j++ && k++' is executed first. The result of this expression is 0 (-1 && -1 && 0 = 0). Now the expression is 0 || 2 which evaluates to 1 (because OR operator always gives 1 except for '0 || 0' combination- for which it gives 0). So the value of m is 1. The values of other variables are also incremented by 1.

44. main()

```
{
char *p;
printf("%d %d ",sizeof(*p),sizeof(p));
}
```

Answer: 1 2

Explanation: The sizeof() operator gives the number of bytes taken by its operand. P is a character pointer, which needs one byte for storing its value (a character). Hence sizeof(*p) gives a value of 1. Since it needs two bytes to store the address of the character pointer sizeof(p) gives 2.

45. main()

```
{
int i=3;
switch(i)
{
default:printf("zero");
case 1: printf("one");
break;
case 2:printf("two");
break;
case 3: printf("three");
break;
}
}
```

Answer : Three

Explanation: The default case can be placed anywhere inside the loop. It is executed only when all other cases doesn't match.

46. main()

```
{
printf("%x",-1<<4);
```

```
}
```

Answer: fff0

Explanation: -1 is internally represented as all 1's. When left shifted four times the least significant 4 bits are filled with 0's.The %x format specifier specifies that the integer value be printed as a hexadecimal value.

47. main()

```
{
char string[]="Hello World";
display(string);
}
void display(char *string)
{
printf("%s",string);
}
```

Answer: Compiler Error: Type mismatch in redeclaration of function display

Explanation: In third line, when the function display is encountered, the compiler doesn't know anything about the function display. It assumes the arguments and return types to be integers, (which is the default type). When it sees the actual function display, the arguments and type contradicts with what it has assumed previously. Hence a compile time error occurs.

48. main()

```
{
int c=- -2;
printf("c=%d",c);
}
```

Answer: c=2;

Explanation: Here unary minus (or negation) operator is used twice. Same maths rules applies, ie. minus * minus= plus.

Note: However you cannot give like --2. Because -- operator can only be applied to variables as a decrement operator (eg., i--). 2 is a constant and not a variable.

49. #define int char

```
main()
{
int i=65;
```

```
printf("sizeof(i)=%d",sizeof(i));
}
```

Answer: sizeof(i)=1

Explanation: Since the #define replaces the string int by the macro char

50. main()

```
{
int i=10;
i=!i>14;
Printf ("i=%d",i);
}
```

Answer: i=0

Explanation: In the expression !i>14 , NOT (!) operator has more precedence than ‘ >’ symbol. ! is a unary logical operator. !i (!10) is 0 (not of true is false). 0>14 is false (zero).

51. #include<stdio.h>

```
main()
{
char s[]={'a','b','c','\n','c','\0'};
char *p,*str,*str1;
p=&s[3];
str=p;
str1=s;
printf("%d",++*p + ++*str1-32);
}
```

Answer: 77

Explanation: p is pointing to character '\n'. str1 is pointing to character 'a' ++*p. "p is pointing to '\n' and that is incremented by one." the ASCII value of '\n' is 10, which is then incremented to 11. The value of ++*p is 11. ++*str1, str1 is pointing to 'a' that is incremented by 1 and it becomes 'b'. ASCII value of 'b' is 98.

Now performing (11 + 98 – 32), we get 77("M"); So we get the output 77 :: "M" (Ascii is 77).

52. #include<stdio.h>

```
main()
{
int a[2][2][2] = { {10,2,3,4}, {5,6,7,8} };
int *p,*q;
```

```
p=&a[2][2][2];
*q=***a;
printf("%d----%d",*p,*q);
}
```

Answer: SomeGarbageValue---1

Explanation: p=&a[2][2][2] you declare only two 2D arrays, but you are trying to access the third 2D(which you are not declared) it will print garbage values.

*q=***a starting address of a is assigned integer pointer. Now q is pointing to starting address of a. If you print *q, it will print first element of 3D array.

53. #include<stdio.h>

```
main()
{
struct xx
{
int x=3;
char name[]="hello";
};
struct xx *s;
printf("%d",s->x);
printf("%s",s->name);
}
```

Answer: Compiler Error

Explanation: You should not initialize variables in declaration

54. #include<stdio.h>

```
main()

{
struct xx
{
int x;
struct yy
{
char s;
struct xx *p;
};
struct yy *q;
```

```
};
}
```

Answer: Compiler Error

Explanation: The structure yy is nested within structure xx. Hence, the elements are of yy are to be accessed through the instance of structure xx, which needs an instance of yy to be known. If the instance is created after defining the structure the compiler will not know about the instance relative to xx. Hence for nested structure yy you have to declare member.

55. main()

```
{
printf("\nab");
printf("\bsi");
printf("\rha");
}
```

Answer: hai

Explanation:
\n - newline
\b - backspace
\r - linefeed

56. main()

```
{
int i=5;
printf("%d%d%d%d%d%d",i++,i--,++i,--i,i);
}
```

Answer: 45545

Explanation: The arguments in a function call are pushed into the stack from left to right. The evaluation is by popping out from the stack. And the evaluation is from right to left, hence the result.

57. #define square(x) x*x

```
main()
{
int i;
i = 64/square(4);
printf("%d",i);
}
```

Answer: 64

Explanation: the macro call square(4) will substituted by 4*4 so the expression becomes i = 64/4*4 . Since / and * has equal priority the expression will be evaluated as (64/4)*4 i.e. 16*4 = 64

58. main()

```
{
char *p="hai friends",*p1;
p1=p;
while(*p!='\0') ++*p++;
printf("%s %s",p,p1);
}
```

Answer: ibj!gsjfoet

Explanation: ++*p++ will be parse in the given order

_ *p that is value at the location currently pointed by p will be taken
_ ++*p the retrieved value will be incremented
_ when; is encountered the location will be incremented that is p++ will be executed Hence, in the while loop initial value pointed by p is ‘h’, which is changed to ‘i’ by executing ++*p and pointer moves to point, ‘a’ which is similarly changed to ‘b’ and so on. Similarly blank space is converted to ‘!’. Thus, we obtain value in p becomes “ibj!gsjfoet” and since p reaches ‘\0’ and p1 points to p thus p1doesnot print anything.

59. #include <stdio.h>

```
#define a 10
main()
{
#define a 50
printf("%d",a);
}
```

Answer: 50

Explanation: The preprocessor directives can be redefined anywhere in the program. So the most recently assigned value will be taken.

60. #define clrscr() 100

```
main()
{
clrscr();
```

```
printf("%d\n",clrscr());
}
```

Answer: 100

Explanation: Preprocessor executes as a seperate pass before the execution of the compiler. So textual replacement of clrscr() to 100 occurs. The input program to compiler looks like this :

```
main()
{
100;
printf("%d\n",100);
}
```

Note: 100; is an executable statement but with no action. So it doesn't give any problem

61. main()

```
{
41printf("%p",main);
}8
```

Answer: Some address will be printed.

Explanation: Function names are just addresses (just like array names are addresses). main() is also a function. So the address of function main will be printed. %p in printf specifies that the argument is an address. They are printed as hexadecimal numbers.

62. main()

```
{
clrscr();
}
clrscr();
```

Answer: No output/error

Explanation: The first clrscr() occurs inside a function. So it becomes a function call. In the second clrscr(); is a function declaration (because it is not inside any function).

63. enum colors {BLACK,BLUE,GREEN}

```
main()
{
printf("%d..%d..%d",BLACK,BLUE,GREEN);
return(1);
}
```

Answer: 0..1..2

Explanation: enum assigns numbers starting from 0, if not explicitly defined.

64. void main()

```
{
char far *farther,*farthest;
printf("%d..%d",sizeof(farther),sizeof(farthest));
}
```

Answer: 4..2

Explanation: The second pointer is of char type and not a far pointer

65. main()

```
{
int i=400,j=300;
printf("%d..%d");
}
```

Answer: 400..300

Explanation: printf takes the values of the first two assignments of the program. Any number of printf's may be given. All of them take only the first two values. If more number of assignments given in the program,then printf will take garbage values.

66. main()

```
{
char *p;
p="Hello";
```

```
printf("%c\n",*&*p);
}
```

Answer: H

Explanation: * is a dereference operator & is a reference operator. They can be applied any number of times provided it is meaningful. Here p points to the first character in the string "Hello". *p dereferences it and so its value is H. Again & references it to an address and * dereferences it to the value H.

67. main()

```
{
int i=1;
while (i<=5)
{
printf("%d",i);
if (i>2)
goto here;
i++;
}
}
fun()
{
here:
printf("PP");
}
```

Answer: Compiler error: Undefined label 'here' in function main

Explanation: Labels have functions scope, in other words the scope of the labels is limited to functions. The label 'here' is available in function fun() Hence it is not visible in function main.

68. main()

```
{
static char names[5][20]={"pascal","ada","cobol","fortran","perl"};
int i;
char *t;
t=names[3];
names[3]=names[4];
names[4]=t;
for (i=0;i<=4;i++)
```

```
printf("%s",names[i]);
}
```

Answer: Compiler error: Lvalue required in function main

Explanation: Array names are pointer constants. So it cannot be modified.

69. void main()

```
{
int i=5;
printf("%d",i++ + ++i);
}
```

Answer: Output Cannot be predicted exactly.

Explanation: Side effects are involved in the evaluation of i

70. void main()

```
{
int i=5;
printf("%d",i+++++i);
}
```

Answer: Compiler Error

Explanation: The expression i+++++i is parsed as i ++ ++ + i which is an illegal combination of operators.

71. #include<stdio.h>

```
main()
{
int i=1,j=2;
switch(i)
{
case 1: printf("GOOD");
break;
case j: printf("BAD");
```

```
break;

}

}
```

Answer: Compiler Error: Constant expression required in function main.

Explanation: The case statement can have only constant expressions (this implies that we cannot use variable names directly so an error).

Note: Enumerated types can be used in case statements.

72. main()

```
{
int i;
printf("%d",scanf("%d",&i)); // value 10 is given as input here

}
```

Answer: 1

Explanation: Scanf returns number of items successfully read and not 1/0. Here 10 is given as input which should have been scanned successfully. So number of items read is 1.

73. #define f(g,g2) g##g2

```
main()
{
int var12=100;
printf("%d",f(var,12));
}
```

Answer: 100

74. main()

```
{
int i=0;
for(;i++;printf("%d",i)) ;
```

```
printf("%d",i);
}
```

Answer: 1

Explanation: before entering into the for loop the checking condition is "evaluated". Here it evaluates to 0 (false) and comes out of the loop, and i is incremented (note the semicolon after the for loop).

75. #include<stdio.h>

```
main()
{
char s[]={'a','b','c','\n','c','\0'};
char *p,*str,*str1;
p=&s[3];
str=p;
str1=s;
printf("%d",++*p + ++*str1-32);
}
```

Answer: M

Explanation: p is pointing to character '\n'.str1 is pointing to character 'a' ++*p "p is pointing to '\n' and that is incremented by one." the ASCII value of '\n' is 10. then it is incremented to 11. the value of ++*p is 11. ++*str1 "str1 is pointing to 'a' that is incremented by 1 and it becomes 'b'. ASCII value of 'b' is 98. Both 11 and 98 is added and result is subtracted from 32. i.e. (11+98-32)=77("M");

76. #include<stdio.h>

```
main()
{
struct xx
{
int x=3;
char name[]="hello";
};
struct xx *s=malloc(sizeof(struct xx));
printf("%d",s->x);
printf("%s",s->name);
}
```

Answer: Compiler Error

Explanation: Initialization should not be done for structure members inside the structure declaration

77. #include<stdio.h>

```
main()
{
struct xx
{
int x;
struct yy
{
char s;
struct xx *p;
};
struct yy *q;
};
}
```

Answer: Compiler Error

Explanation: in the end of nested structure yy a member have to be declared.

78. main()

```
{
extern int i;
i=20;
printf("%d",sizeof(i));
}
```

Answer: Linker error: undefined symbol '_i'.

Explanation: extern declaration specifies that the variable i is defined somewhere else. The compiler passes the external variable to be resolved by the linker. So compiler doesn't find an error. During linking the linker searches for the definition of i. Since it is not found the linker flags an error.

79. main()

```
{
printf("%d", out);
}
int out=100;
```

Answer: Compiler error: undefined symbol out in function main.

Explanation: The rule is that a variable is available for use from the point of declaration. Even though a is a global variable, it is not available for main. Hence an error.

80. main()

```
{
extern out;
printf("%d", out);
}
int out=100;
```

Answer: 100

Explanation: This is the correct way of writing the previous program.

81. main()

```
{
show();
}
void show()
{
printf("I'm the greatest");
}
```

Answer: Compier error: Type mismatch in redeclaration of show.

Explanation: When the compiler sees the function show it doesn't know anything about it. So the default return type (ie, int) is assumed. But when compiler sees the actual definition of show mismatch occurs since it is declared as void. Hence the error.

The solutions are as follows:
1. declare void show() in main() .

2. define show() before main().
3. declare extern void show() before the use of show().

82. main( )

```
{
int a[2][3][2] = {{{2,4},{7,8},{3,4}},{{2,2},{2,3},{3,4}}};
printf("%u %u %u %d \n",a,*a,**a,***a);
printf("%u %u %u %d \n",a+1,*a+1,**a+1,***a+1);
}
```

Answer:

100, 100, 100, 2
114, 104, 102, 3

Explanation: The given array is a 3-D one. It can also be viewed as a 1-D array.

2	4	7	8	3	4	2	2	2	3	3	4

100 102 104 106 108 110 112 114 116 118 120 122

thus, for the first printf statement a, *a, **a give address of first element. since the indirection ***a gives the value. Hence, the first line of the output.

for the second printf a+1 increases in the third dimension thus points to value at 114, *a+1 increments in second dimension thus points to 104, **a +1 increments the first dimension thus points to 102 and ***a+1 first gets the value at first location and then increments it by 1. Hence, the output.

83. main( )

```
{
int a[ ] = {10,20,30,40,50},j,*p;
for(j=0; j<5; j++)
{
printf("%d" ,*a);
a++;
}
p = a;
for(j=0; j<5; j++)
{
printf("%d " ,*p);
p++;
```

```
}
}
```

Answer: Compiler error: lvalue required.

Explanation: Error is in line with statement a++. The operand must be an lvalue and may be of any of scalar type for the any operator, array name only when subscripted is an lvalue. Simply array name is a non modifiable lvalue.

84. main( )

```
{
static int a[ ] = {0,1,2,3,4};
int *p[ ] = {a,a+1,a+2,a+3,a+4};
int **ptr = p;
ptr++;
printf("\n %d %d %d", ptr-p, *ptr-a, **ptr);
*ptr++;
printf("\n %d %d %d", ptr-p, *ptr-a, **ptr);
*++ptr;
printf("\n %d %d %d", ptr-p, *ptr-a, **ptr);
++*ptr;
printf("\n %d %d %d", ptr-p, *ptr-a, **ptr);
}
```

Answer:

111
222
333
344

Explanation: Let us consider the array and the two pointers with some address

a

0	1	2	3	4

100 102 104 106 108

p

100	102	104	106	108

1000 1002 1004 1006 1008

ptr

1000

2000

After execution of the instruction ptr++ value in ptr becomes 1002, if scaling factor for integer is 2 bytes. Now ptr – p is value in ptr –starting location of array p, (1002 – 1000) / (scaling factor) = 1,

*ptr – a = value at address pointed by ptr – starting value of array a, 1002 has a value 102 so the value is (102 – 100)/(scaling factor) = 1, **ptr is the value stored in the location pointed by the pointer of ptr = value pointed by value pointed by 1002 = value pointed by 102 = 1. Hence the output of the firs printf is 1, 1, 1.

After execution of *ptr++ increments value of the value in ptr by scaling factor, so it becomes1004. Hence, the outputs for the second printf are ptr – p = 2, *ptr – a = 2, **ptr = 2.

After execution of *++ptr increments value of the value in ptr by scaling factor, so it becomes1004. Hence, the outputs for the third printf are ptr – p = 3, *ptr – a = 3, **ptr = 3.

After execution of ++*ptr value in ptr remains the same, the value pointed by the value is incremented by the scaling factor. So the value in array p at location 1006 changes from 106 10 108,. Hence, the outputs for the fourth printf are ptr – p = 1006 – 1000 = 3, *ptr – a = 108 – 100 = 4, **ptr = 4.

85. What is dangling pointer in c?

If any pointer is pointing the memory address of any variable but after some variable has deleted from that memory location while pointer is still pointing such memory location. Such pointer is known as dangling pointer and this problem is known as dangling pointer problem.

86. What are merits and demerits of array in c?

Merits:
(a) We can easily access each element of array.
(b) Not necessity to declare too many variables.
(c) Array elements are stored in continuous memory location.

Demerits:
(a) Wastage of memory space. We cannot change size of array at the run time.
(b) It can store only similar type of data

87. Where are the auto variables stored?

Auto variables are stored in main memory and their default value is a garbage value.

88. Why Preincrement operator is faster than Postincrement?

Evaluation of any expression is from left to right. Preincrement is faster because it doesn't need to save the current value for next instruction whereas Postincrement needs to saves current value to be incremented after execution of current instruction.

89. Difference between arrays and linked list?

Major differences between arrays and linked lists are: (i) In array consecutive elements are stored in consecutive memory locations whereas in linked list it not so. (ii) In array address of next element is consecutive and whereas in linked list it is specified in the address part of each node.(iii) Linked List makes better use of memory than arrays.(iv) Insertion or deletion of an element in array is difficult than insertion or deletion in linked list

90. What is the use of typedef?

(i)It increases the portability.
(ii) It simplify the complex declaration and improve readability of the program.

91. What are library Functions?

Library Functions are predefined functions and stored in .lib files.

92. What is a structure?

Structure is a collection of heterogeneous (i.e. related data items which can be of different types) held together to a single unit. The data items enclosed within a structure are called its members which may be of data type int, float, char, array etc.

93. What is a pointer?

Pointer is a variable that contains address of another variable in the memory. Pointers are quite useful in creation of linked data structures (such as linked lst, trees graphs), managing object allocated memory dynamically, optimize the program to execute faster and use less memory.

94. What are the techniques you use for debugging?

(i)Using compiler's features
(ii)Read The Fine Module
(iii)printf( ) debugging
(iv)Code grinding
(v)Assertion

95. What are macros? What are its advantages and disadvantages?

Macro is a Pre-processor.Major advantage of using the macro is to increase the speed of the execution of the program.
Major disadvantage of the macros are:
(i) No type checking is performed in macro. This may cause error.
(ii) A macro call may cause unexpected results.

96. What is difference between Structure and Unions?

(i) In structure every member has its own memory whereas in union its members share the same member space.
(ii) In structure, it is possible to initialize all the members at the same time which is not possible in case of union.
(iii) A structure requires more space than union(for the same type of members).
(iv) In union different interpretations of the same memory space are possible which is not so in case of structures.

97. What are the advantages of using Unions?

(i) Efficient use of memory as it it does not demand memory space for its all members rather it require memory space for its largest member only.
(ii) Same memory space can be interpreted differently for different members of the union.

98. What is the difference between ordinary variable and pointer in C?

An ordinary variable is like a container it can hold any value and we can change the value of ordinary variable at a time throughout the program .A pointer is a variable that stores the address of another Variable.

99. What are segment and offset addresses?
When paging technique is performed, the page will breaks into segments and its sequence is

said to be segments and its width can be said as offset. In short,segment is a physical address and offset is logical address.

100. When should a type cast be used?

There are two situations in which to use a type cast. The first use is to change the type of an operand to an arithmetic operation so that the operation will be performed properly.
The second case is to cast pointer types to and from void * in order to interface with functions that expect or return void pointers. For example, the following line type casts the return value of the call to malloc() to be a pointer to a foo structure.
struct foo *p = (struct foo *) malloc(sizeof(struct foo));

101. What is the difference between %d and %*d in c language?

%d give the original value of the variable and %*d give the address of the variable.

eg:-int a=10,b=20;
printf("%d%d",a,b);
printf("%*d%*d",a,b);
Result is 10 20 1775 1775 .Here 1775 is the starting address of the memory allocation for the integer.a and b having same address because of contagious memory allocation.

102. How does a C program come to know about command line arguments?
When we execute our C program, operating system loads the program into memory. In case of DOS, it first loads 256 bytes into memory, called program segment prefix. This contains file tables,environment segment, and command line information. When we compile the C program the compiler inserts additional code that parses the command, assigning it to the argv array, making the arguments easily accessible within our C program.

103. How are pointer variables initialized?

Pointer variable are initialized by one of the following two ways
- Static memory allocation
- Dynamic memory allocation

104. What is modular programming?

If a program is large, it is subdivided into a number of smaller
programs that are called modules or subprograms. If a complex

problem is solved using more modules, this approach is known as modular programming

105. Where does global, static, local, register variables and C Program instructions get stored?

Global , static, local : In main memory
Register variable: In registers
C program : In main memory.

106. Where are the auto variables stored?

Auto variables are stored in main memory and their default value is a garbage value.

107. What is an lvalue?

An lvalue is an expression to which a value can be assigned. The lvalue expression is located on the left side of an assignment statement, whereas an rvalue is located on the right side of an assignment statement. Each assignment statement must have an lvalue and an rvalue. The lvalue expression must reference a storable variable in memory. It cannot be a constant

108. What is an argument? Differentiate between formal arguments and actual arguments?

An argument is an entity used to pass the data from calling function to the called function. Formal arguments are the arguments available in the function definition. They are preceded by their own data types. Actual arguments are available in the function call.

109. When is a switch statement better than multiple if statements?

A switch statement is generally best to use when you have more than two conditional expressions based on a single variable of numeric type.

110. Differentiate between a linker and linkage?

A linker converts an object code into an executable code by linking together the necessary build in functions. The form and place of declaration where the variable is declared in a program determine the
linkage of variable.

111. Define Operator, Operand, and Expression in 'C'?

Operators are symbols which take one or more operands or expressions and perform arithmetic or logical computations.
Operands are variables or expressions which are used in operators to evaluate the expression.
Combination of operands and operators form an expression.

112. What will be the result of the following code?

```
#define TRUE 0 // some code
while(TRUE)
{
   // some code
}
```

Answer: This will not go into the loop as TRUE is defined as 0.

113. What will be printed as the result of the operation below:

```
main()
{
   int a=0;
   if(a==0)
      printf("Cisco Systemsn");
      printf("Cisco Systemsn");
}
```

Answer: Two lines with "Cisco Systems" will be printed.

114. Do you know pragma directives in c?

Pragma is implementation specific directive i.e each pragma directive has different implementation rule and use. If compiler does not recognize particular pragma it simply ignore that pragma statement without showing any error or warning message and execute the whole program assuming this pragma statement is not present.

115. Predict the output or error

```
main()
{
clrscr();
}
clrscr();
```

Ans:No output/error
Explanation:The first clrscr() occurs inside a function. So it becomes a function call. In the second clrscr(); is a function declaration (because it is not inside any function).

116. Predict the output or error

```
enum colors {BLACK,BLUE,GREEN}
 main()
{

 printf("%d..%d..%d",BLACK,BLUE,GREEN);

 return(1);
}
```

Answer: 0..1..2
Explanation: enum assigns numbers starting from 0, if not explicitly defined.

117. Predict the output or error

```
 main()
{
int i;
printf("%d",scanf("%d",&i));  // value 10 is given as input here
}
```

Answer:1
Explanation: Scanf returns number of items successfully read and not 1/0. Here 10 is given as input which should have been scanned successfully. So number of items read is 1.

118. what will be the position of the file marker?

a: fseek(ptr,0,SEEK_SET);
b: fseek(ptr,0,SEEK_CUR);

Ans: a: The SEEK_SET sets the file position marker to the starting of the file.
b: The SEEK_CUR sets the file position marker to the current position of the file.

119. Predict the output or error

```
main()
        {
        main();
        }
```

Ans: Runtime error : Stack overflow.
Explanation: main function calls itself again and again. Each time the function is called its return address is stored in the call stack. Since there is no condition to terminate the function call, the call stack overflows at runtime. So it terminates the program and results in an error.

120. Predict the output or error

```
main()
{
         int i=5,j=6,z;
         printf("%d",i+++j);
        }
```

Answer:11

Explanation:the expression i+++j is treated as (i++ + j)

121. Predict the output or error

```
main()
        {
        int k=1;
        printf("%d==1 is ""%s",k,k==1?"TRUE":"FALSE");
        }
```

Ans: 1==1 is TRUE
Explanation: When two strings are placed together (or separated by white-space) they are concatenated (this is called as "stringization" operation). So the string is as if it is given as "%d==1 is %s". The conditional operator(?:) evaluates to "TRUE".

122. What is use of void data type?

Void is an empty data type normally used as a return type in C/C++, C#, Java functions/methods to declare that no value will be return by the function.
The another used of void is to declare the pointer in C/C++ where It is not sure what data type is addressed by the pointer.

123. four type of scope in c:

Block scope.
Function scope.
File scope.
Program scope.

124. Tell any five properties of auto variables?

auto variables are defined inside a function. A variable declared inside the function without storage class name is, by default, an auto variable. These functions are declared on the stack. The stack provides temporary storage.

125. What is automatic type promotion in c?

In c if two operands are of different data type in a binary operation then before performing any operation compiler will automatically convert the operand of lower data type to higher data type .This phenomenon is known as automatic type conversion. For example:
int a=10,c;
float b=5.5f;
c=a+b;
Here a int variable while b is float variable. So before performing addition operation value of the variable a (Lower data type) will automatically convert into float constant (higher data type) then it will perform addition operation.

126. What are differences between sizeof operator and strlen function?

sizeof is keyword of c which can find size of a string constant including null character but strlen is function which has been defined string.h and can find number of characters in a string excluding null character.

127. What is command line argument?

Getting the arguments from command prompt in c is known as command line arguments. In c main function has three arguments.
They are:
Argument counter
Argument vector
Environment vector

128. void main(){

```
   int x=5,y=10,z=15,val;
   val=sum(x,(y=0,z=0,y),z);
   clrscr();
   printf("%d",val);
   getch();
}
sum(int x,int y,int z){
   return x+y+z;
}
```

Answer:20
Explanation: In the above program comma after Y=0 &Z=0 are behaving as operator.

129. what is nested structure?

A structure is a collection of one or more variables, possibly of different data types, grouped together under a single name for convenient handling. Structures can contain other structures as members; in other words, structures can nest.

130. What is slack byte in structure?

To store any type of data in structure there is minimum fixed byte which must be reserved by memory. This minimum byte is known as word boundary. Word boundary depends upon machine. TURBO C is based on 8086 microprocessor which has two byte word boundary. So any data type reserves at least two byte space.

131.What is prototype of printf function?

Prototype of printf function is:
int printf(const char *format ,…)

132.What is difference between declaration and definition?
During declaration we just specify the type and no memory is allocated to the variable. But during the
definition an initial value is assigned and memory is allocated to the variable.

133. What is function recursion?

When a function of body calls the same function then it is called as 'recursive function.'
Example:

```
Recursion()
{
   printf("Recursion !");
   Recursion();
}
```

134. What is self referential structure ?

A self-referential structure is one of the data structures which refer to the pointer to (points) to another structure of the same type.

135. What is far pointer?

The pointer which can point or access whole the residence memory of RAM i.e. which can access all 16 segments is known as far pointer.

136. What is pascal and cdecl keyword in c language?

There are two types of parameters passing conventions in c:
1. pascal: In this style function name should (not necessary) in the uppercase .First parameter of function call is passed to the first parameter of function definition and so on.
2. cdecl: In this style function name can be both in the upper case or lower case. First parameter of function call is passed to the last parameter of function definition. It is default parameter passing convention.

137. What is use of #pragma inline directive in c language?

#pragma inline only tells the compiler that source code of program contain inline assembly language code .In c we can write assembly language program with help of asm keyword.

138. What is the meaning of multilevel pointers in c?

A pointer is pointer to another pointer which can be pointer to others pointers and so on is known as multilevel pointers. We can have any level of pointers.

139. What is huge pointer in c?

The pointer which can point or access whole the residence memory of RAM i.e. which can access all the 16 segments is known as huge pointer.

140. Is it possible to rename any function in c?

Yes, we can rename any function using typedef keyword. It is useful when function declaration is too complex and we have to give any simple name or if we have to create more numbers of function of the same type.

141. Do you know, what is the meaning and use of static keyword in c?

Keyword static is used for declaring static variables in c. This modifier is used with all data types like int, float, double, array, pointer, structure, function etc.

142. What is difference between .com program and .exe program?

Both .com and .exe program are executable program but .com program execute faster than .exe program. All drivers are .com program. .com file has higher preference than .exe For example:

143. Difference between TSR and TSO program

TSO means terminate but stay outside. It is that program, which release the main memory after the execution of the program. Example ms paint, notepad, turbo c compilers etc.

TSR means terminate but stay residence .It is those program, which after the execution of the program does not release the RAM (main memory).e.g. antivirus.

144. Describe turbo c compiler?

Turbo c is an IDE of c programming language created by Borland. Turbo C 3.0 is based on MS DOS operation system. It is one of the most popular c compilers. It uses 8086 microprocessor which is 16 bit microprocessor. It has 20 address buses and 16 data bus. Its word length is two byte.

145. Out of fgets() and gets() which function is safe to use and why?

fgets() is safer than gets(), because we can specify a maximum input length. Neither one is completely safe, because the compiler can't prove that programmer won't overflow the buffer he pass to fgets ().

146. Difference between strdup and strcpy?

Both copy a string. strcpy wants a buffer to copy into. strdup allocates a buffer using malloc().
Unlike strcpy(), strdup() is not specified by ANSI .

147. Differentiate between a for loop and a while loop? What are it uses?

For executing a set of statements fixed number of times we use for loop while when the number of
iterations to be performed is not known in advance we use while loop.

148. What is storage class? What are the different storage classes in C?
Storage class is an attribute that changes the behavior of a variable. It controls the lifetime, scope and linkage. The storage classes in c are auto, register, and extern, static, typedef.

149. What are the uses of a pointer?

(i)It is used to access array elements
(ii)It is used for dynamic memory allocation.
(iii)It is used in Call by reference
(iv)It is used in data structures like trees, graph, linked list etc.

150.In header files whether functions are declared or defined?

Functions are declared within header file. That is function prototypes exist in a header file,not function bodies. They are defined in library (lib).

151. Difference between pass by reference and pass by value?

Pass by reference passes a pointer to the value. This allows the callee to modify the variable directly.Pass by value gives a copy of the value to the callee. This allows the callee to modify the value without modifying the variable. (In other words, the callee simply cannot modify the variable, since it lacks a reference to it.)

152. What are enumerations?

They are a list of named integer-valued constants. Example:enum color { black , orange=4,yellow, green, blue, violet };This declaration defines the symbols “black”, “orange”, “yellow”, etc. to have the values “1,” “4,” “5,” … etc. The difference between an enumeration and a macro is that the enum actually declares a type, and therefore can be type checked.

153. Are pointers integer?

No, pointers are not integers. A pointer is an address. It is a positive number.

154. What is static memory allocation?

Compiler allocates memory space for a declared variable. By using the address of operator, the reserved address is obtained and this address is assigned to a pointer variable. This way of assigning pointer value to a pointer variable at compilation time is known as static memory allocation.

155. What is dynamic memory allocation?

A dynamic memory allocation uses functions such as malloc() or calloc() to get memory dynamically. If these functions are used to get memory dynamically and the values returned by these function are assigned to pointer variables, such a way of allocating memory at run time is known as dynamic memory allocation.

156. What modular programming?

If a program is large, it is subdivided into a number of smaller programs that are called modules or subprograms. If a complex problem is solved using more modules, this approach is known as modular programming

157. What is a function?

A large program is subdivided into a number of smaller programs or subprograms. Each subprogram specifies one or more actions to be performed for the larger program. Such sub programs are called functions.

158. Difference between formal argument and actual argument?
Formal arguments are the arguments available in the function definition. They are preceded by their own data type. Actual arguments are available in the function call. These arguments are given as constants or variables or expressions to pass the values to the function.

159. what are C tokens?

There are six classes of tokens: identifier, keywords, constants, string literals, operators and other separators.

160. What are C identifiers?

These are names given to various programming element such as variables, function, arrays.It is a combination of letter, digit and underscore.It should begin with letter. Backspace is not allowed.

161. Difference between syntax vs logical error?

Syntax Error
These involves validation of syntax of language.
compiler prints diagnostic message.

Logical Error
logical error are caused by an incorrect algorithm or by a statement mistyped in such a way that it doesn't violet syntax of language.
difficult to find.

162. What are the facilities provided by preprocessor?
file inclusion
substitution facility
conditional compilation

163.What do the functions atoi(), itoa() and gcvt() do?

atoi() is a macro that converts integer to character.
itoa() It converts an integer to string
gcvt() It converts a floating point number to string

164. What is FILE?

FILE is a predefined data type. It is defined in stdio.h file.

165. What is a file?

A file is a region of storage in hard disks or in auxiliary storage devices.It contains bytes of information .It is not a data type.

C++ Interview Questions

1. What is C++?

Released in 1985, C++ is an object-oriented programming language created by Bjarne Stroustrup. C++ maintains almost all aspects of the C language, while simplifying memory management and adding several features - including a new datatype known as a class (you will learn more about these later) - to allow object-oriented programming. C++ maintains the features of C which allowed for low-level memory access but also gives the programmer new tools to simplify memory management.
C++ used for:
C++ is a powerful general-purpose programming language. It can be used to create small programs or large applications. It can be used to make CGI scripts or console-only DOS programs. C++ allows you to create programs to do almost anything you need to do. The creator of C++, Bjarne Stroustrup, has put together a partial list of applications written in C++.

2. How do you find out if a linked-list has an end? (i.e. the list is not a cycle)
You can find out by using 2 pointers. One of them goes 2 nodes each time. The second one goes at 1 nodes each time. If there is a cycle, the one that goes 2 nodes each time will eventually meet the one that goes slower. If that is the case, then you will know the linked-list is a cycle.

3. What is the difference between realloc() and free()?
The free subroutine frees a block of memory previously allocated by the malloc subroutine. Undefined results occur if the Pointer parameter is not a valid pointer. If the Pointer parameter is a null value, no action will occur. The realloc subroutine changes the size of the block of memory pointed to by the Pointer parameter to the number of bytes specified by the Size parameter and returns a new pointer to the block. The pointer specified by the Pointer parameter must have been created with the malloc, calloc, or realloc subroutines and not been deallocated with the free or realloc subroutines. Undefined results occur if the Pointer parameter is not a valid pointer.

4. Base class has some virtual method and derived class has a method with the same name. If we initialize the base class pointer with derived object, calling of that virtual method will result in which method being called?

a. Base method
b. Derived method
Ans. B

5. What is function overloading and operator overloading?
Function overloading: C++ enables several functions of the same name to be defined, as long as these functions have different sets of parameters (at least as far as their types are concerned). This capability is called function overloading. When an overloaded function is called, the C++ compiler selects the proper function by examining the number, types and order of the arguments in the call. Function overloading is commonly used to create several functions of the same name that perform similar tasks but on different data types.

Operator overloading allows existing C++ operators to be redefined so that they work on objects of user-defined classes. Overloaded operators are syntactic sugar for equivalent function calls. They form a pleasant facade that doesn't add anything fundamental to the language (but they can improve understandability and reduce maintenance costs).

6. What are the advantages of inheritance?
It permits code reusability. Reusability saves time in program development. It encourages the reuse of proven and debugged high-quality software, thus reducing problem after a system becomes functional.

7.What is the difference between declaration and definition?
The declaration tells the compiler that at some later point we plan to present the definition of this declaration.
E.g.: void stars () //function declaration
The definition contains the actual implementation.
E.g.: void stars () // declarator
{
for(int j=10; j > =0; j--) //function body
cout << *;
cout << endl;
}

8. How do you write a function that can reverse a linked-list?

```
void reverselist(void)
{
if(head==0)
return;
if(head->next==0)
return;
if(head->next==tail)
{
head->next = 0;
tail->next = head;
}
```

```
else
{
node* pre = head;
node* cur = head->next;
node* curnext = cur->next;
head->next = 0;
cur-> next = head;
for(; curnext!=0; )
{
cur->next = pre;
pre = cur;
cur = curnext;
curnext = curnext->next;
}

curnext->next = cur;
}
}
```

9. What do you mean by inline function?

The idea behind inline functions is to insert the code of a called function at the point where the function is called. If done carefully, this can improve the application's performance in exchange for increased compile time and possibly (but not always) an increase in the size of the generated binary executables.

10. Write a program that ask for user input from 5 to 9 then calculate the average

```
#include "iostream.h"
int main() {
int MAX = 4;
int total = 0;
int average;
int numb;
for (int i=0; i<MAX; i++) {
cout << "Please enter your input between 5 and 9: ";
cin >> numb;
while ( numb<5 || numb>9) {
cout << "Invalid input, please re-enter: ";
cin >> numb;
}
total = total + numb;
}
average = total/MAX;
cout << "The average number is: " << average << "\n";
return 0;
}
```

11. Write a short code using C++ to print out all odd number from 1 to 100 using a for loop

```
for( unsigned int i = 1; i < = 100; i++ )
if( i & 0x00000001 )
cout << i << \",\";
```

12. What is public, protected, private?

Public, protected and private are three access specifier in C++.
Public data members and member functions are accessible outside the class.
Protected data members and member functions are only available to derived classes.
Private data members and member functions can't be accessed outside the class. However there is an exception can be using friend classes.

13. Tell how to check whether a linked list is circular.

Create two pointers, each set to the start of the list. Update each as follows: while (pointer1) {
pointer1 = pointer1->next;
pointer2 = pointer2->next; if (pointer2) pointer2=pointer2->next;
if (pointer1 == pointer2) {
print (\"circular\n\");
}
}
OK, why does this work?
If a list is circular, at some point pointer2 will wrap around and be either at the item just before pointer1, or the item before that. Either way, it's either 1 or 2 jumps until they meet.

14. What is virtual constructors/destructors?

Virtual destructors:If an object (with a non-virtual destructor) is destroyed explicitly by applying the delete operator to a base-class pointer to the object, the base-class destructor function (matching the pointer type) is called on the object.
There is a simple solution to this problem declare a virtual base-class destructor. This makes all derived-class destructors virtual even though they don't have the same name as the base-class destructor. Now, if the object in the hierarchy is destroyed explicitly by applying the delete operator to a base-class pointer to a derived-class object, the destructor for the appropriate class is called. Virtual constructor: Constructors cannot be virtual. Declaring a constructor as a virtual function is a syntax error.
Virtual destructors: If an object (with a non-virtual destructor) is destroyed explicitly by applying the delete operator to a base-class pointer to the object, the base-class destructor function (matching the pointer type) is called on the object.
There is a simple solution to this problem – declare a virtual base-class destructor. This makes all derived-class destructors virtual even though they don't have the same name as the base-class destructor. Now, if the object in the hierarchy is destroyed explicitly by applying the delete operator to a base-class pointer to a derived-class object, the destructor for the appropriate class is called.

Virtual constructor: Constructors cannot be virtual. Declaring a constructor as a virtual function is a syntax error.

15. Does c++ support multilevel and multiple inheritance?
Yes.

16. What are the advantages of inheritance?
• It permits code reusability.
• Reusability saves time in program development.
• It encourages the reuse of proven and debugged high-quality software, thus reducing problem after a system becomes functional.

17. What is the difference between declaration and definition?
The declaration tells the compiler that at some later point we plan to present the definition of this declaration.
E.g.: void stars () //function declaration
The definition contains the actual implementation.
E.g.: void stars () // declarator
{
for(int j=10; j>=0; j--) //function body
cout<<"*";
cout<<endl;
}

18. What is the difference between an array and a list?

Array is collection of homogeneous elements. List is collection of heterogeneous elements. For Array memory allocated is static and continuous. For List memory allocated is dynamic and Random.
Array: User need not have to keep in track of next memory allocation.
List: User has to keep in Track of next location where memory is allocated.
Array uses direct access of stored members, list uses sequencial access for members.
/With Array you have direct access to memory position 5
Object x = a[5]; // x takes directly a reference to 5th element of array
//With the list you have to cross all previous nodes in order to get the 5th node:
list mylist;
list::iterator it;
for(it = list.begin() ; it != list.end() ; it++)
{
if(i==5)
{
x = *it;
break;
}

```
i++;
}
```

19. What is a template?

Templates allow to create generic functions that admit any data type as parameters and return value without having to overload the function with all the possible data types. Until certain point they fulfill the functionality of a macro. Its prototype is any of the two following ones: template <class indetifier> function_declaration; template <typename indetifier> function_declaration;

The only difference between both prototypes is the use of keyword class or typename, its use is indistinct since both expressions have exactly the same meaning and behave exactly the same way.

20. Define a constructor - What it is and how it might be called (2 methods).

Constructor is a member function of the class, with the name of the function being the same as the class name. It also specifies how the object should be initialized.

Ways of calling constructor:

1) Implicitly: automatically by complier when an object is created.

2) Calling the constructors explicitly is possible, but it makes the code unverifiable.

```
class Point2D{
int x; int y;
public Point2D() : x(0) , y(0) {} //default (no argument) constructor
};
main(){
Point2D MyPoint; // Implicit Constructor call. In order to allocate memory on stack, the default constructor is implicitly called.
Point2D * pPoint = new Point2D(); // Explicit Constructor call. In order to allocate memory on HEAP we call the default constructor.
```

You have two pairs: new() and delete() and another pair : alloc() and free().

21. Explain differences between eg. new() and malloc()

1.) “new and delete” are preprocessors while “malloc() and free()” are functions. [we dont use brackets will calling new or delete].

2.) no need of allocate the memory while using “new” but in “malloc()” we have to use “sizeof()”.

3.) “new” will initlize the new memory to 0 but “malloc()” gives random value in the new alloted memory location [better to use calloc()]

new() allocates continous space for the object instace malloc() allocates distributed space.

new() is castless, meaning that allocates memory for this specific type,

malloc(), calloc() allocate space for void * that is cated to the specific class type pointer.

22. What is the difference between class and structure?

Structure: Initially (in C) a structure was used to bundle different type of data types together to perform a particular functionality. But C++ extended the structure to contain functions

also. The major difference is that all declarations inside a structure are by default public. Class: Class is a successor of Structure. By default all the members inside the class are private.

23. What is RTTI?
Runtime type identification (RTTI) lets you find the dynamic type of an object when you have only a pointer or a reference to the base type. RTTI is the official way in standard C++ to discover the type of an object and to convert the type of a pointer or reference (that is, dynamic typing).

24.What is encapsulation?
Packaging an object's variables within its methods is called encapsulation.

25.**Explain term "Polymorphism"** and give an example using eg. SHAPE object: If I have a base class SHAPE, how would I define DRAW methods for two objects CIRCLE and SQUARE"Polymorphism": A phenomenon which enables an object to react differently to the same function call. in C++ it is attained by using a keyword virtual
Example

```
public class SHAPE
{
public virtual void SHAPE::DRAW()=0;
}
```

Note here the function DRAW() is pure virtual which means the sub classes must implement the DRAW() method and SHAPE cannot be instatiated

```
public class CIRCLE::public SHAPE
{
public void CIRCLE::DRAW()
{
// TODO drawing circle
}
}
public class SQUARE::public SHAPE
{
public void SQUARE::DRAW()
{
// TODO drawing square
}
}
```

now from the user class the calls would be like globally

```
SHAPE *newShape;
```

When user action is to draw

```
public void MENU::OnClickDrawCircle(){
```

```
newShape = new CIRCLE();
}

public void MENU::OnClickDrawCircle(){
newShape = new SQUARE();
}

the when user actually draws
public void CANVAS::OnMouseOperations(){
newShape->DRAW();
}
class SHAPE{
public virtual Draw() = 0; //abstract class with a pure virtual method
};
class CIRCLE{
public int r;
public virtual Draw() { this->drawCircle(0,0,r); }
};

class SQURE
public int a;
public virtual Draw() { this->drawRectangular(0,0,a,a); }
};
```

Each object is driven down from SHAPE implementing Draw() function in its own way.

26. What is an object?

Object is a software bundle of variables and related methods. Objects have state and behavior.

27. How can you tell what shell you are running on UNIX system?

You can do the Echo $RANDOM. It will return a undefined variable if you are from the C-Shell, just a return prompt if you are from the Bourne shell, and a 5 digit random numbers if you are from the Korn shell. You could also do a ps -l and look for the shell with the highest PID.

28. What do you mean by inheritance?

Inheritance is the process of creating new classes, called derived classes, from existing classes or base classes. The derived class inherits all the capabilities of the base class, but can add embellishments and refinements of its own.

29.Describe PRIVATE, PROTECTED and PUBLIC – the differences and give examples.

```
class Point2D{
int x; int y;
public int color;
protected bool pinned;
public Point2D() : x(0) , y(0) {} //default (no argument) constructor
};
Point2D MyPoint;
```

You cannot directly access private data members when they are declared (implicitly) private:

```
MyPoint.x = 5; // Compiler will issue a compile ERROR
//Nor yoy can see them:
int x_dim = MyPoint.x; // Compiler will issue a compile ERROR
```

On the other hand, you can assign and read the public data members:

```
MyPoint.color = 255; // no problem
int col = MyPoint.color; // no problem
```

With protected data members you can read them but not write them: MyPoint.pinned = true; // Compiler will issue a compile ERROR
bool isPinned = MyPoint.pinned; // no problem

30. What is namespace?

Namespaces allow us to group a set of global classes, objects and/or functions under a name. To say it somehow, they serve to split the global scope in sub-scopes known as namespaces. The form to use namespaces is:
namespace identifier { namespace-body }
Where identifier is any valid identifier and namespace-body is the set of classes, objects and functions that are included within the namespace. For example:
namespace general { int a, b; } In this case, a and b are normal variables integrated within the general namespace. In order to access to these variables from outside the namespace we have to use the scope operator ::. For example, to access the previous variables we would have to put:
general::a general::b
The functionality of namespaces is specially useful in case that there is a possibility that a global object or function can have the same name than another one, causing a redefinition error.

31. What is a COPY CONSTRUCTOR and when is it called?

A copy constructor is a method that accepts an object of the same class and copies it's data members to the object on the left part of assignment:

```
class Point2D{
int x; int y;
public int color;
protected bool pinned;
public Point2D() : x(0) , y(0) {} //default (no argument) constructor
public Point2D( const Point2D & ) ;
};
Point2D::Point2D( const Point2D & p )
```

```
{
this->x = p.x;
this->y = p.y;
this->color = p.color;
this->pinned = p.pinned;
}

main(){
Point2D MyPoint;
MyPoint.color = 345;
Point2D AnotherPoint = Point2D( MyPoint ); // now AnotherPoint has color = 345
```

32. What is Boyce Codd Normal form?

A relation schema R is in BCNF with respect to a set F of functional dependencies if for all functional dependencies in F+ of the form a-> , where a and b is a subset of R, at least one of the following holds:

* a- > b is a trivial functional dependency (b is a subset of a)
* a is a superkey for schema R

33. What is virtual class and friend class?

Friend classes are used when two or more classes are designed to work together and need access to each other's implementation in ways that the rest of the world shouldn't be allowed to have. In other words, they help keep private things private. For instance, it may be desirable for class DatabaseCursor to have more privilege to the internals of class Database than main() has.

34. What is the word you will use when defining a function in base class to allow this function to be a polimorphic function?

virtual

35. What do you mean by binding of data and functions?

Encapsulation.

36. What are 2 ways of exporting a function from a DLL?

1. Taking a reference to the function from the DLL instance.
2. Using the DLL ’s Type Library

37. What is the difference between an object and a class?

Classes and objects are separate but related concepts. Every object belongs to a class and every class contains one or more related objects.
- A Class is static. All of the attributes of a class are fixed before, during, and after the execution of a program. The attributes of a class don't change.
- The class to which an object belongs is also (usually) static. If a particular object belongs to a certain class at the time that it is created then it almost certainly will still belong to that class right up until the time that it is destroyed.
- An Object on the other hand has a limited lifespan. Objects are created and eventually destroyed. Also during that lifetime, the attributes of the object may undergo significant change.

38. What is a class?

Class is a user-defined data type in C++. It can be created to solve a particular kind of problem. After creation the user need not know the specifics of the working of a class.

39. What is friend function?

As the name suggests, the function acts as a friend to a class. As a friend of a class, it can access its private and protected members. A friend function is not a member of the class. But it must be listed in the class definition.

40. Which recursive sorting technique always makes recursive calls to sort subarrays that are about half size of the original array?

Mergesort always makes recursive calls to sort subarrays that are about half size of the original array, resulting in O(n log n) time.

41. What is abstraction?

Abstraction is of the process of hiding unwanted details from the user.

42. What are virtual functions?

A virtual function allows derived classes to replace the implementation provided by the base class. The compiler makes sure the replacement is always called whenever the object in

question is actually of the derived class, even if the object is accessed by a base pointer rather than a derived pointer. This allows algorithms in the base class to be replaced in the derived class, even if users don't know about the derived class.

43.What is the difference between an external iterator and an internal iterator? Describe an advantage of an external iterator.

An internal iterator is implemented with member functions of the class that has items to step through. .An external iterator is implemented as a separate class that can be "attach" to the object that has items to step through. .An external iterator has the advantage that many difference iterators can be active simultaneously on the same object.

44. What is a scope resolution operator?
A scope resolution operator (::), can be used to define the member functions of a class outside the class.

45. What do you mean by pure virtual functions?

A pure virtual member function is a member function that the base class forces derived classes to provide. Normally these member functions have no implementation. Pure virtual functions are equated to zero.
class Shape { public: virtual void draw() = 0; };

46. What is polymorphism? Explain with an example?

"Poly" means "many" and "morph" means "form". Polymorphism is the ability of an object (or reference) to assume (be replaced by) or become many different forms of object. Example: function overloading, function overriding, virtual functions. Another example can be a plus '+' sign, used for adding two integers or for using it to concatenate two strings.

47.What's the output of the following program? Why?

```
#include <stdio.h>
main()
{
typedef union
{
int a;
char b[10];
float c;
```

```
}
Union;
Union x,y = {100};
x.a = 50;
strcpy(x.b,\"hello\");
x.c = 21.50;
printf(\"Union x : %d %s %f \n\",x.a,x.b,x.c );
printf(\"Union y :%d %s%f \n\",y.a,y.b,y.c);
}
```

Given inputs X, Y, Z and operations | and & (meaning bitwise OR and AND, respectively) What is output equal to in
output = (X & Y) | (X & Z) | (Y & Z)

48. Why arrays are usually processed with for loop?

The real power of arrays comes from their facility of using an index variable to traverse the array, accessing each element with the same expression a[i]. All the is needed to make this work is a iterated statement in which the variable i serves as a counter, incrementing from 0 to a.length -1. That is exactly what a loop does.

49. What is an HTML tag?

An HTML tag is a syntactical construct in the HTML language that abbreviates specific instructions to be executed when the HTML script is loaded into a Web browser. It is like a method in Java, a function in C++, a procedure in Pascal, or a subroutine in FORTRAN.

50.Explain which of the following declarations will compile and what will be constant - a pointer or the value pointed at: * const char *
* char const *
* char * const

51. What problems might the following macro bring to the application?

```
#define sq(x) x*x
```

52. Anything wrong with this code?

```
T *p = new T[10];
delete p;
```

Everything is correct, Only the first element of the array will be deleted", The entire array will be deleted, but only the first element destructor will be called.

53. Anything wrong with this code?

T *p = 0;
delete p;
Yes, the program will crash in an attempt to delete a null pointer.

54. How do you decide which integer type to use?

It depends on our requirement. When we are required an integer to be stored in 1 byte (means less than or equal to 255) we use short int, for 2 bytes we use int, for 8 bytes we use long int. A char is for 1-byte integers, a short is for 2-byte integers, an int is generally a 2-byte or 4-byte integer (though not necessarily), a long is a 4-byte integer, and a long long is a 8-byte integer.

55. What does extern mean in a function declaration?

Using extern in a function declaration we can make a function such that it can used outside the file in which it is defined.
An extern variable, function definition, or declaration also makes the described variable or function usable by the succeeding part of the current source file. This declaration does not replace the definition. The declaration is used to describe the variable that is externally defined.
If a declaration for an identifier already exists at file scope, any extern declaration of the same identifier found within a block refers to that same object. If no other declaration for the identifier exists at file scope, the identifier has external linkage.

56. What can I safely assume about the initial values of variables which are not explicitly initialized?

It depends on complier which may assign any garbage value to a variable if it is not initialized.

57. What is the difference between char a[] = "string"; and char *p = "string";?

In the first case 6 bytes are allocated to the variable a which is fixed, where as in the second case if *p is assigned to some other value the allocate memory can change.

58. What's the auto keyword good for?
Not much. It declares an object with automatic storage duration. Which means the object will be destroyed at the end of the objects scope. All variables in functions that are not declared as static and not dynamically allocated have automatic storage duration by default.
For example
int main()
{
int a; //this is the same as writing "auto int a;"
}
Local variables occur within a scope; they are "local" to a function. They are often called automatic variables because they automatically come into being when the scope is entered and automatically go away when the scope closes. The keyword auto makes this explicit, but local variables default to auto auto auto auto so it is never necessary to declare something as an auto auto auto auto.

59. What is the difference between char a[] = "string"; and char *p = "string"; ?

a[] = "string";
char *p = "string";
The difference is this:
p is pointing to a constant string, you can never safely say
p[3]='x';
however you can always say a[3]='x';
char a[]="string"; - character array initialization.
char *p="string" ; - non-const pointer to a const-string.(this is permitted only in the case of char pointer in C++ to preserve backward compatibility with C.)

60. How do I declare an array of N pointers to functions returning pointers to functions returning pointers to characters?

If you want the code to be even slightly readable, you will use typedefs.
typedef char* (*functiontype_one)(void);
typedef functiontype_one (*functiontype_two)(void);
functiontype_two myarray[N]; //assuming N is a const integral

61. What does extern mean in a function declaration?
It tells the compiler that a variable or a function exists, even if the compiler hasn't yet seen it in the file currently being compiled. This variable or function may be defined in another file or further down in the current file.

62. How do I initialize a pointer to a function?

This is the way to initialize a pointer to a function

```
void fun(int a)
{
}
void main()
{
void (*fp)(int);
fp=fun;
fp(1);
}
```

63. How do you link a C++ program to C functions?

By using the extern "C" linkage specification around the C function declarations.

64. Explain the scope resolution operator.

It permits a program to reference an identifier in the global scope that has been hidden by another identifier with the same name in the local scope.

65. What are the differences between a C++ struct and C++ class?

The default member and base-class access specifier are different.

66. How many ways are there to initialize an int with a constant?

Two. There are two formats for initializers in C++ as shown in the example that follows. The first format uses the traditional C notation. The second format uses constructor notation.

```
int foo = 123;
int bar (123);
```

67. How does throwing and catching exceptions differ from using setjmp and longjmp?

The throw operation calls the destructors for automatic objects instantiated since entry to the try block.

68. What is a default constructor?

Default constructor WITH arguments class B { public: B (int m = 0) : n (m) {} int n; }; int main(int argc, char *argv[]) { B b; return 0; }

69. What is a conversion constructor?
A constructor that accepts one argument of a different type.

70. What is the difference between a copy constructor and an overloaded assignment operator?

A copy constructor constructs a new object by using the content of the argument object. An overloaded assignment operator assigns the contents of an existing object to another existing object of the same class.

71. When should you use multiple inheritance?

There are three acceptable answers: "Never," "Rarely," and "When the problem domain cannot be accurately modeled any other way."

72. Explain the ISA and HASA class relationships. How would you implement each in a class design?

A specialized class "is" a specialization of another class and, therefore, has the ISA relationship with the other class. An Employee ISA Person. This relationship is best implemented with inheritance. Employee is derived from Person. A class may have an instance of another class. For example, an employee "has" a salary, therefore the Employee class has the HASA relationship with the Salary class. This relationship is best implemented by embedding an object of the Salary class in the Employee class.

73. When is a template a better solution than a base class?

When you are designing a generic class to contain or otherwise manage objects of other types, when the format and behavior of those other types are unimportant to their containment or management, and particularly when those other types are unknown (thus, the generosity) to the designer of the container or manager class.

74. What is a mutable member?

One that can be modified by the class even when the object of the class or the member function doing the modification is const.

75. What is an explicit constructor?

A conversion constructor declared with the explicit keyword. The compiler does not use an explicit constructor to implement an implied conversion of types. It's purpose is reserved explicitly for construction.

76. What is the Standard Template Library (STL)?

A library of container templates approved by the ANSI committee for inclusion in the standard C++ specification. A programmer who then launches into a discussion of the generic programming model, iterators, allocators, algorithms, and such, has a higher than average understanding of the new technology that STL brings to C++ programming.

77. Describe run-time type identification.

The ability to determine at run time the type of an object by using the typeid operator or the dynamic cast operator.

78. What problem does the namespace feature solve?

Multiple providers of libraries might use common global identifiers causing a name collision when an application tries to link with two or more such libraries. The namespace feature surrounds a library's external declarations with a unique namespace that eliminates the potential for those collisions.
This solution assumes that two library vendors don't use the same namespace identifier, of course.

79. Are there any new intrinsic (built-in) data types?

Yes. The ANSI committee added the bool intrinsic type and its true and false value keywords.

80. Will the following program execute?

```
void main()
{
void *vptr = (void *) malloc(sizeof(void));
vptr++;
```

```
}
```
It will throw an error, as arithmetic operations cannot be performed on void pointers.

81. For the following C program

```
#define AREA(x)(3.14*x*x)
main()
{
float r1=6.25,r2=2.5,a;
a=AREA(r1);
printf("\n Area of the circle is %f", a);
a=AREA(r2);
printf("\n Area of the circle is %f", a);
}
```
What is the output?
Ans. Area of the circle is 122.656250
Area of the circle is 19.625000

82. void main()

```
{
int d=5;
printf("%f",d);
}
```
Ans: Undefined

```
83. void main()
{
int i;
for(i=1;i<4,i++)
switch(i)
case 1: printf("%d",i);break;
{
case 2:printf("%d",i);break;
case 3:printf("%d",i);break;
}
switch(i) case 4:printf("%d",i);
}
```
Ans: 1,2,3,4

```
84.void main()
{
char *s="\12345s\n";
printf("%d",sizeof(s));
```

}
Ans: 6

85.void main()
{
unsigned i=1; /* unsigned char k= -1 => k=255; */
signed j=-1; /* char k= -1 => k=65535 */
/* unsigned or signed int k= -1 =>k=65535 */
if(i<j)
printf("less");
else
if(i>j)
printf("greater");
else
if(i==j)
printf("equal");
}
Ans: less</j)

86. void main()
{
float j;
j=1000*1000;
printf("%f",j);
}
1. 1000000
2. Overflow
3. Error
4. None
Ans: 4

87. How do you declare an array of N pointers to functions returning pointers to functions returning pointers to characters?

Ans: The first part of this question can be answered in at least three ways:
1. char *(*(*a[N])())();
2. Build the declaration up incrementally, using typedefs:
typedef char *pc; /* pointer to char */
typedef pc fpc(); /* function returning pointer to char */
typedef fpc *pfpc; /* pointer to above */
typedef pfpc fpfpc(); /* function returning... */
typedef fpfpc *pfpfpc; /* pointer to... */
pfpfpc a[N]; /* array of... */
3. Use the cdecl program, which turns English into C and vice versa:
cdecl> declare a as array of pointer to function returning pointer to function returning pointer

to char
char *(*(*a[])())()

88.What is a modifier?

A modifier, also called a modifying function is a member function that changes the value of at least one data member. In other words, an operation that modifies the state of an object. Modifiers are also known as 'mutators'.

89.What is an accessor?

An accessor is a class operation that does not modify the state of an object. The accessor functions need to be declared as const operations

90.Differentiate between a template class and class template.

Template class: A generic definition or a parameterized class not instantiated until the client provides the needed information. It's jargon for plain templates.

Class template: A class template specifies how individual classes can be constructed much like the way class specifies how individual objects can be constructed. It's jargon for plain classes

91.When does a name clash occur?

A name clash occurs when a name is defined in more than one place. For example., two different class libraries could give two different classes the same name. If you try to use many class libraries at the same time, there is a fair chance that you will be unable to compile or link the program because of name clashes.

92.What is a dangling pointer?

A dangling pointer arises when you use the address of an object after its lifetime is over. This may occur in situations like returning addresses of the automatic variables from a function or using the address of the memory block after it is freed.

93.Differentiate between the message and method.

Message:

Objects communicate by sending messages to each other.
A message is sent to invoke a method

Method:

Provides response to a message.
It is an implementation of an operation.

94.What is an adaptor class or Wrapper class?

A class that has no functionality of its own. Its member functions hide the use of a third party software component or an object with the non-compatible interface or a non-object-oriented implementation.

95.What is a Null object?

It is an object of some class whose purpose is to indicate that a real object of that class does not exist. One common use for a null object is a return value from a member function that is supposed to return an object with some specified properties but cannot find such an object.

96.What is class invariant?
A class invariant is a condition that defines all valid states for an object. It is a logical condition to ensure the correct working of a class. Class invariants must hold when an object is created, and they must be preserved under all operations of the class. In particular all class invariants are both preconditions and post-conditions for all operations or member functions of the class.

97.What do you mean by Stack unwinding?

It is a process during exception handling when the destructor is called for all local objects between the place where the exception was thrown and where it is caught.

98.What are proxy objects?

Objects that stand for other objects are called proxy objects or surrogates.

99.Name some pure object oriented languages.

Smalltalk, Java, Eiffel, Sather.

100.What is an orthogonal base class?

If two base classes have no overlapping methods or data they are said to be independent of, or orthogonal to each other. Orthogonal in the sense means that two classes operate in different dimensions and do not interfere with each other in any way. The same derived class may inherit such classes with no difficulty.

101.What is the difference between Mutex and Binary semaphore?

semaphore is used to synchronize processes. where as mutex is used to provide synchronization between threads running in the same process

102.What is destructor?

Destructor usually deletes any extra resources allocated by the object.

103.What are C++ storage classes?

auto
register
static
extern
auto:the default. Variables are automatically created and initialized when they are defined and are destroyed at the end of the block containing their definition. They are not visible outside that block
register:a type of auto variable. a suggestion to the compiler to use a CPU register for performance
static:a variable that is known only in the function that contains its definition but is never destroyed and retains its value between calls to that function. It exists from the time the program begins execution
extern:a static variable whose definition and placement is determined when all object and library modules are combined (linked) to form the executable code file. It can be visible outside the file where it is defined.

104.What is reference ?

reference is a name that acts as an alias, or alternative name, for a previously defined variable or an object.prepending variable with "&" symbol makes it as reference.
for example:
int a;
int &b = a;

105.What are the defining traits of an object-oriented language?

The defining traits of an object-oriented langauge are:
encapsulation
inheritance
polymorphism

106.What is Quadratic Probing?

The Performance problem encountered by linear probing is caused by the cluster buildup That occurs as a result of the probing sequence. Quadratic probing uses a different sequence to avoid primary clustering.

107.What is the chaining?

The Chaining technique basically looks at the hash table as an array of pointers to linked lists. Each slot in the hash table is either empty or simply consists of a pointer to a linked list. You resolve collisions by adding the elements that hash to the same slot to the linked list to which that slot points. At the same time, deletions are easy, You simply delete elements from the linked list.

108.What is the Hash Function?

The hash function is an important part of the hashing technique. This function is used to transform the keys into table addresses. The hash function we choose should be easy to compute and should be able to transform the keys into integers in the range 0 to TR-1. Because most of the commonly used hash functions are based on arithmetic operations, We should convert the keys to numbers on which arithmetic operations can be performed

109.What is an Visualizations?

The visualization is the basically a way of presentation ,Its just a fancy name for the diagrams, pictures, screen shots, prototypes, and any other visual representations created to help through and design the graphical user interface of your product.

110.What is virtual inheritance?

Inheritance is a basically can be private , public, or virtual. With virtual inheritance there is only one copy of each object even if the object appears more than once in the hierarchy.

111.What is multithreading

Multithreading is defined as :It is the task of creating a new thread of execution within an existing process rather than starting a new process to begin a function. It is the ability of an operating system to concurrently run programs that have been divided into subcomponents, or threads.

112.What is the use of using?

Using is bassically a namespace scope. Its directive used to declare the accessibility of identifiers declared within a namespace scope.

113.What is the use of exception handling?

Exception handling is bassically used to detect exceptions becouse it can be taken a corresponding action

114.What is EOF?

EOF bassically stands for End of File, It is used to check for the end of file when a file is being read.

115.Define the parameterized macros?

Parameterized macros are use for the parameters . It is the one which consist of template with insertion points for the addition of parameters.

116.What is overflow error?

Overflow error basically a type of arithmatic errors.It's caused by the result of an arithmetic operation being greater than the actual space provided by the system.

117.What is a nested class? Why can it be useful?
Nested classes basically useful for organizing code and controlling access and dependencies. Nested classes obey access rules just like other parts of a class do.and that class is a class enclosed within the scope of another class.

118.What are the disadvantages of C++?

a)It's not pure object oriented programming language.

b)Its a Platform dependent
c)C++ does not give excellant graphics as compare to java.
d)Its Not case sensitive.
e)C++ have Less features as comapred to Java& C#.
f)Its Not applicable in web enviornment.
g)Does not provide very strong type-checking.
h)c++ code is easily prone to errors related to data types, their conversions.
i)Does not provide efficient means for garbage collection.
j)No built in support for threads

119.What is an iterator?

An iterator is a bassically a type of object that represents a stream of data. It is Unlike a sequence, an iterator can only provide the next item. The for-in statement uses iterators to control the loop, and iterators can also be used in many other contexts

120.What is the Auto Storage Class?

Auto Storage Class is bassically the default. Variables are automatically created and initialized, When they are defined and are destroyed at the end of the block containing their definition. They are not visible outside that block.

121.What is callback function?

Callback function is the type of pointer for a function

122.What is the use of tellg ()?

tellg () provides a information about the current position of input/get pointer.

123.What is the use of tellp ()?

tellp ()use for the poitner postion :Its provides the current position of output/put pointer

124.Define the generic programming?

Generic Programmng is type of method. The method in which generic types are used as arguments in algorithms for different data types and data structures is called generic programming.

125.What is the use of Microsoft foundation class library?

The Microsoft Foundation Class Library also called as A Microsoft Foundation Classes or MFC. It is basically a library that wraps portions of the Windows API in C++ classes, and including functionality that enables them to use a default application framework. Classes are defined for many of the handle-managed Windows objects and also for predefined windows and common controls. MFC library would help us reduce the code and development time.

Java Interview Questions

1.What is JVM?

The Java interpreter along with the runtime environment required to run the Java application in called as Java virtual machine(JVM)

2. What is the most important feature of Java?

Java is a platform independent language.

3. What do you mean by platform independence?

Platform independence means that we can write and compile the java code in one platform (eg Windows) and can execute the class in any other supported platform eg (Linux,Solaris,etc).

4. What is the difference between a JDK and a JVM?

JDK is Java Development Kit which is for development purpose and it includes execution environment also. But JVM is purely a run time environment and hence you will not be able to compile your source files using a JVM.

5. What is the base class of all classes?

java.lang.Object

6. What are the access modifiers in Java?

There are 3 access modifiers. Public, protected and private, and the default one if no identifier is specified is called friendly, but programmer cannot specify the friendly identifier explicitly.

7. What is are packages?

A package is a collection of related classes and interfaces providing access protection and namespace management.

8. What is meant by Inheritance and what are its advantages?

Inheritance is the process of inheriting all the features from a class. The advantages of inheritance are reusability of code and accessibility of variables and methods of the super class by subclasses.

9. What is the difference between superclass and subclass?

A super class is a class that is inherited whereas sub class is a class that does the inheriting.

10. What is an abstract class?

An abstract class is a class designed with implementation gaps for subclasses to fill in and is deliberately incomplete.

11. What are the states associated in the thread?

Thread contains ready, running, waiting and dead states.

12. What is synchronization?

Synchronization is the mechanism that ensures that only one thread is accessed the resources at a time.

13. What is deadlock?

When two threads are waiting each other and can't precede the program is said to be deadlock.

14. What is an applet?

Applet is a dynamic and interactive program that runs inside a web page displayed by a java capable browser

15. What is the lifecycle of an applet?

init() method - Can be called when an applet is first loaded
start() method - Can be called each time an applet is started.
paint() method - Can be called when the applet is minimized or maximized.
stop() method - Can be used when the browser moves off the applet's page.
destroy() method - Can be called when the browser is finished with the applet.

16. How do you set security in applets?

using setSecurityManager() method

17. What is a layout manager and what are different types of layout managers available in java AWT?

A layout manager is an object that is used to organize components in a container. The different layouts are available are FlowLayout, BorderLayout, CardLayout, GridLayout and GridBagLayout

18. What is JDBC?

JDBC is a set of Java API for executing SQL statements. This API consists of a set of classes and interfaces to enable programs to write pure Java Database applications.

19. What are drivers available?

-a) JDBC-ODBC Bridge driver b) Native API Partly-Java driver
c) JDBC-Net Pure Java driver d) Native-Protocol Pure Java driver

20. What is stored procedure?

Stored procedure is a group of SQL statements that forms a logical unit and performs a particular task. Stored Procedures are used to encapsulate a set of operations or queries to execute on database. Stored procedures can be compiled and executed with different parameters and results and may have any combination of input/output parameters.

21. 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.

22. Why there are no global variables in Java?

Global variables are globally accessible. Java does not support globally accessible variables due to following reasons:
1)The global variables breaks the referential transparency
2)Global variables creates collisions in namespace.

23. What are Encapsulation, Inheritance and Polymorphism?

Encapsulation is the mechanism that binds together code and data it manipulates and keeps both safe from outside interference and misuse. Inheritance is the process by which one object acquires the properties of another object. Polymorphism is the feature that allows one interface to be used for general class actions.

24. What is the use of bin and lib in JDK?

Bin contains all tools such as javac, appletviewer, awt tool, etc., whereas lib contains API and all packages.

25. What is method overloading and method overriding?

Method overloading: When a method in a class having the same method name with different arguments is said to be method overloading. Method overriding : When a method in a class having the same method name with same arguments is said to be method overriding.

26. What is the difference between this() and super()?

this() can be used to invoke a constructor of the same class whereas super() can be used to invoke a super class constructor.

27. What is Domain Naming Service(DNS)?

It is very difficult to remember a set of numbers(IP address) to connect to the Internet. The Domain Naming Service(DNS) is used to overcome this problem. It maps one particular IP address to a string of characters. For example, www. mascom. com implies com is the domain name reserved for US commercial sites, moscom is the name of the company and www is the name of the specific computer, which is mascom's server.

28. What is URL?

URL stands for Uniform Resource Locator and it points to resource files on the Internet. URL has four components: http://www. address. com:80/index.html, where http - protocol name, address - IP address or host name, 80 - port number and index.html - file path.

29. What is RMI and steps involved in developing an RMI object?

Remote Method Invocation (RMI) allows java object that executes on one machine and to invoke the method of a Java object to execute on another machine. The steps involved in developing an RMI object are: a) Define the interfaces b) Implementing these interfaces c) Compile the interfaces and their implementations with the java compiler d) Compile the server implementation with RMI compiler e) Run the RMI registry f) Run the application.

30. What is RMI architecture?

RMI architecture consists of four layers and each layer performs specific functions: a) Application layer - contains the actual object definition. b) Proxy layer - consists of stub and skeleton. c) Remote Reference layer - gets the stream of bytes from the transport layer and sends it to the proxy layer. d) Transportation layer - responsible for handling the actual machine-to-machine communication.

31. What is a Java Bean?

A Java Bean is a software component that has been designed to be reusable in a variety of different environments.

32. What are checked exceptions?

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

33. What are runtime exceptions?

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.

34. What is the difference between error and an exception?

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.).

35. What is the purpose of finalization?

The purpose of finalization is to give an unreachable object the opportunity to perform any cleanup processing before the object is garbage collected. For example, closing a opened file, closing a opened database Connection.

36. What is the difference between yielding and sleeping?

When a task invokes its yield() method, it returns to the ready state. When a task invokes its sleep() method, it returns to the waiting state.

37. What is the difference between preemptive scheduling and time slicing?

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.

38. What is mutable object and immutable object?

If a object value is changeable then we can call it as Mutable object. (Ex., StringBuffer, …) If you are not allowed to change the value of an object, it is immutable object. (Ex., String, Integer, Float, …)

39. What is the purpose of Void class?

The Void class is an uninstantiable placeholder class to hold a reference to the Class object representing the primitive Java type void.

40. What is JIT and its use?

Really, just a very fast compiler… In this incarnation, pretty much a one-pass compiler — no offline computations. So you can't look at the whole method, rank the expressions according to which ones are re-used the most, and then generate code. In theory terms, it's an on-line problem.

41. What is nested class?

If all the methods of a inner class is static then it is a nested class.

42. What is HashMap and Map?

Map is Interface and Hashmap is class that implements that.

43. What are different types of access modifiers?

public: Any thing declared as public can be accessed from anywhere. private: Any thing declared as private can't be seen outside of its class. protected: Any thing declared as protected can be accessed by classes in the same package and subclasses in the other packages. default modifier : Can be accessed only to classes in the same package.

44. What is the difference between Reader/Writer and InputStream/Output Stream?

The Reader/Writer class is character-oriented and the InputStream/OutputStream class is byte-oriented.

45. What is servlet?

Servlets are modules that extend request/response-oriented servers, such as java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.

46. What is Constructor?

A constructor is a special method whose task is to initialize the object of its class.
It is special because its name is the same as the class name.
They do not have return types, not even void and therefore they cannot return values.
They cannot be inherited, though a derived class can call the base class constructor.
Constructor is invoked whenever an object of its associated class is created.

47. What is an Iterator ?

The Iterator interface is used to step through the elements of a Collection.
Iterators let you process each element of a Collection.
Iterators are a generic way to go through all the elements of a Collection no matter how it is organized.
Iterator is an Interface implemented a different way for every Collection.

48. What is the List interface?

The List interface provides support for ordered collections of objects.
Lists may contain duplicate elements.

49. What is memory leak?

A memory leak is where an unreferenced object that will never be used again still hangs around in memory and doesnt get garbage collected.

50. What is the difference between the prefix and postfix forms of the ++ operator?

The prefix form performs the increment operation and returns the value of the increment operation. The postfix form returns the current value all of the expression and then performs the increment operation on that value.

51. What is the difference between a constructor and a method?

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.

52. What will happen to the Exception object after exception handling?

Exception object will be garbage collected.

53. Difference between static and dynamic class loading.

Static class loading: The process of loading a class using new operator is called static class loading. Dynamic class loading: The process of loading a class at runtime is called dynamic class loading.
Dynamic class loading can be done by using Class.forName(….).newInstance().

54. Explain the Common use of EJB

The EJBs can be used to incorporate business logic in a web-centric application.
The EJBs can be used to integrate business processes in Business-to-business (B2B) e-commerce applications.In Enterprise Application Integration applications, EJBs can be used to house processing and mapping between different applications.

55. What is JSP?

JSP is a technology that returns dynamic content to the Web client using HTML, XML and JAVA elements. JSP page looks like a HTML page but is a servlet. It contains Presentation logic and business logic of a web application.

56. What is the purpose of apache tomcat?

Apache server is a standalone server that is used to test servlets and create JSP pages. It is free and open source that is integrated in the Apache web server. It is fast, reliable server to configure the applications but it is hard to install. It is a servlet container that includes tools to configure and manage the server to run the applications. It can also be configured by editing XML configuration files.

57. Where pragma is used?

Pragma is used inside the servlets in the header with a certain value. The value is of no-cache that tells that a servlets is acting as a proxy and it has to forward request. Pragma directives allow the compiler to use machine and operating system features while keeping the overall functionality with the Java language. These are different for different compilers.

58. Briefly explain daemon thread.

Daemon thread is a low priority thread which runs in the background performs garbage collection operation for the java runtime system.

59. What is a native method?

A native method is a method that is implemented in a language other than Java.

60. Explain different way of using thread?

A Java thread could be implemented by using Runnable interface or by extending the Thread class. The Runnable is more advantageous, when you are going for multiple inheritance.

61. What are the two major components of JDBC?

One implementation interface for database manufacturers, the other implementation interface for application and applet writers.

62. What kind of thread is the Garbage collector thread?

It is a daemon thread.

63. What are the different ways to handle exceptions?

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 handle those exceptions.

64. How many objects are created in the following piece of code?

MyClass c1, c2, c3;
c1 = new MyClass ();
c3 = new MyClass ();
Answer: Only 2 objects are created, c1 and c3. The reference c2 is only declared and not initialized.

65.What is UNICODE?

Unicode is used for internal representation of characters and strings and it uses 16 bits to represent each other.

DOT NET Interview Questions

1.What is .NET?

NET is an integral part of many applications running on Windows and provides common functionality for those applications to run. This download is for people who need .NET to run an application on their computer. For developers, the .NET Framework provides a comprehensive and consistent programming model for building applications that have visually stunning user experiences and seamless and secure communication.

2.How many languages .NET is supporting now?

When .NET was introduced it came with several languages.
VB.NET,
C#,
COBOL
and
Perl, etc.

3. What is an IL?

Intermediate Language is also known as MSIL (Microsoft Intermediate Language) or CIL (Common Intermediate Language). All .NET source code is compiled to IL. IL is then converted to machine code at the point where the software is installed, or at run-time by a Just-In-Time (JIT) compiler.

4. What is code access security (CAS)?

Code access security (CAS) is part of the .NET security model that prevents unauthorized access of resources and operations, and restricts the code to perform particular tasks.

5. What is Difference between NameSpace and Assembly?

Assembly is physical grouping of logical units, Namespace, logically groups classes. Namespace can span multiple assembly.

6. Mention the execution process for managed code.

A)Choosing a language compiler
B) Compiling the code to MSIL
C) Compiling MSIL to native code
D) Executing the code.

7. What is Microsoft Intermediate Language (MSIL)?

The .NET Framework is shipped with compilers of all .NET programming languages to develop programs. There are separate compilers for the Visual Basic, C#, and Visual C++ programming languages in .NET Framework. Each .NET compiler produces an intermediate code after compiling the source code. The intermediate code is common for all languages and is understandable only to .NET environment. This intermediate code is known as MSIL.

8. What is managed extensibility framework?

Managed extensibility framework (MEF) is a new library that is introduced as a part of .NET 4.0 and Silverlight 4. It helps in extending your application by providing greater reuse of applications and components. MEF provides a way for host application to consume external extensions without any configuration requirement.

9. Which method do you use to enforce garbage collection in .NET?

The System.GC.Collect() method.

10. What is the difference between int and int32.

There is no difference between int and int32. System.Int32 is a .NET Class and int is an alias name for System.Int32.

11. What are tuples?

Tuple is a fixed-size collection that can have elements of either same or different data types. Similar to arrays, a user must have to specify the size of a tuple at the time of declaration. Tuples are allowed to hold up from 1 to 8 elements and if there are more than 8 elements, then the 8th element can be defined as another tuple. Tuples can be specified as parameter or return type of a method.

12. What is the full form of ADO?

The full form of ADO is ActiveX Data Object.

13. What are the two fundamental objects in ADO.NET?

DataReader and DataSet are the two fundamental objects in ADO.NET.

14. What is the meaning of object pooling?

Object pooling is a concept of storing a pool (group) of objects in memory that can be reused later as needed. Whenever, a new object is required to create, an object from the pool can be allocated for this request; thereby, minimizing the object creation. A pool can also refer to a group of connections and threads. Pooling, therefore, helps in minimizing the use of system resources, improves system scalability, and performance.

15. Mention the namespace that is used to include .NET Data Provider for SQL server in .NET code.

The System.Data.SqlClient namespace.

16. Which architecture does Datasets follow?

Datasets follow the disconnected data architecture.

17. What is the role of the DataSet object in ADO.NET?

One of the major component of ADO.NET is the DataSet object, which always remains disconnected from the database and reduces the load on the database.

18. Which property is used to check whether a DataReader is closed or opened?

The IsClosed property is used to check whether a DataReader is closed or opened. This property returns a true value if a Data Reader is closed, otherwise a false value is returned.

19. Name the method that needs to be invoked on the DataAdapter control to fill the generated DataSet with data?

The Fill() method is used to fill the dataset with data.

20. What are the pre-requisites for connection pooling?

There must be multiple processes to share the same connection describing the same parameters and security settings. The connection string must be identical.

21. Which adapter should you use, if you want to get the data from an Access database?

OleDbDataAdapter is used to get the data from an Access database.

22. What are different types of authentication techniques that are used in connection strings to connect .NET applications with Microsoft SQL Server?

The Windows Authentication option

The SQL Server Authentication option

23. What are the parameters that control most of connection pooling behaviors?

Connect Timeout
Max Pool Size
Min Pool Size
Pooling

24. What is AutoPostBack?

If you want a control to postback automatically when an event is raised, you need to set the AutoPostBack property of the control to True.

25. What is the function of the ViewState property?

The ASP.NET 4.0 introduced a new property called ViewStateMode for the Control class. Now you can enable the view state to an individual control even if the view state for an ASP.NET page is disabled.

26. Which properties are used to bind a DataGridView control?

The DataSource property and the DataMember property are used to bind a DataGridView control.

27. What is the basic difference between ASP and ASP.NET?

The basic difference between ASP and ASP.NET is that ASP is interpreted; whereas, ASP.NET is compiled. This implies that since ASP uses VBScript; therefore, when an ASP page is executed, it is interpreted. On the other hand, ASP.NET uses .NET languages, such as C# and VB.NET, which are compiled to Microsoft Intermediate Language (MSIL).

28. In which event are the controls fully loaded?

Page load event guarantees that all controls are fully loaded. Controls are also accessed in Page_Init events but you will see that view state is not fully loaded during this event

29. How can we identify that the Page is Post Back?

Page object has an "IsPostBack" property, which can be checked to know that is the page posted back.

30. Which is the parent class of the Web server control?

The System.Web.Ul.Control class is the parent class for all Web server controls.

31. What are the advantages of the code-behind feature?

i)Makes code easy to understand and debug by separating application logic from HTML tags
ii)Provides the isolation of effort between graphic designers and software engineers
iii)Removes the problems of browser incompatibility by providing code files to exist on the Web server and supporting Web pages to be compiled on demand.

32. Define a multilingual Web site.

A multilingual Web site serves content in a number of languages. It contains multiple copies for its content and other resources, such as date and time, in different languages.

33. What is IIS? Why is it used?

Internet Information Services (IIS) is created by Microsoft to provide Internet-based services to ASP.NET Web applications. It makes your computer to work as a Web server and provides the functionality to develop and deploy Web applications on the server. IIS handles the request and response cycle on the Web server. It also offers the services of SMTP and FrontPage server extensions. The SMTP is used to send emails and use FrontPage server extensions to get the dynamic features of IIS, such as form handler.

34. How can you register a custom server control to a Web page?

You can register a custom server control to a Web page using the @Register directive.

35. Which ASP.NET objects encapsulate the state of the client and the browser?

The Session object encapsulates the state of the client and browser.

36. Differentiate globalization and localization.

The globalization is a technique to identify the specific part of a Web application that is different for different languages and make separate that portion from the core of the Web application. The localization is a procedure of configuring a Web application to be supported for a specific language or locale.

37. What is ViewState?

The ViewState is a feature used by ASP.NET Web page to store the value of a page and its controls just before posting the page. Once the page is posted, the first task by the page processing is to restore the ViewState to get the values of the controls.

38. Which method is used to force all the validation controls to run?

The Page.Validate() method is used to force all the validation controls to run and to perform validation.

39. What does the Orientation property do in a Menu control?

Orientation property of the Menu control sets the horizontal or vertical display of a menu on a Web page. By default, the orientation is vertical.

40. Differentiate between client-side and server-side validations in Web pages.

Client-side validations take place at the client end with the help of JavaScript and VBScript before the Web page is sent to the server. On the other hand, server-side validations take place at the server end.

41. What is garbage collection?

Garbage collection is a heap-management strategy where a run-time component takes responsibility for managing the lifetime of the memory used by objects. This concept is not new to .NET - Java and many other languages/runtimes have used garbage collection for some time.

42. What is serialization?

Serialization is the process of converting an object into a stream of bytes.Deserialization is the opposite process, i.e. creating an object from a stream of bytes.
Serialization/Deserialization is mostly used to transport objects (e.g. during remoting), or to persist objects (e.g. to a file or database).

43. Where do you add an event handler?
It's the Attributesproperty, the Add function inside that property.
e.g.btnSubmit.Attributes.Add("onMouseOver","someClientCode();")

44. What do you mean by authentication and authorization?

Authentication is the process of validating a user on the credentials(username and password) and authorization performs after authentication. After Authentication a user will be verified for performing the various tasks, It access is limited it is known as authorization.

45. What is portable executable (PE) ?

The file format used for executable programs and for files to be linked together to form executable programs

46. Differences between DLL and EXE?

.exe

1.These are outbound file.
2.Only one .exe file exists per application.
3..Exe cannot be shared with other applications.

.dll
1.These are inbound file .
2.Many .dll files may exists in one application.
3. .dll can be shared with other applications.

47. What is shadowing?

Shadowing is either through scope or through inheritance. Shadowing through inheritance is hiding a method of a base class and providing a new implementation for the same. This is the default when a derived class writes an implementation of a method of base class which is not declared as overridden in the base class. This also serves the purpose of protecting an implementation of a new method against subsequent addition of a method with the same name in the base class.'shadows' keyword is recommended although not necessary since it is the default.

48. What is Method Overriding? How to override a function in C#?

An override method provides a new implementation of a member inherited from a base class. The method overridden by an override declaration is known as the overridden base method. The overridden base method must have the same signature as the override method.
Use the override modifier to modify a method, a property, an indexer, or an event. You cannot override a non-virtual or static method. The overridden base method must be virtual, abstract, or override.

49. Differences between dataset.clone and dataset.copy?

Clone - Copies the structure of the DataSet, including all DataTable schemas, relations, and constraints. Does not copy any data.
Copy - Copies both the structure and data for this DataSet.

50. What is the managed and unmanaged code in .net?

The .NET Framework provides a run-time environment called the Common Language Runtime, which manages the execution of code and provides services that make the development process easier. Compilers and tools expose the runtime's functionality and enable you to write code that benefits from this managed execution environment. Code that you develop with a language compiler that targets the runtime is called managed code; it benefits from features such as cross-language integration, cross-language exception handling, enhanced security, versioning and deployment support, a simplified model for component interaction, and debugging and profiling services.

51. Whats an assembly?

Assemblies are the building blocks of .NET Framework applications; they form the fundamental unit of deployment, version control, reuse, activation scoping, and security permissions. An assembly is a collection of types and resources that are built to work together and form a logical unit of functionality. An assembly provides the common language runtime with the information it needs to be aware of type implementations. To the runtime, a type does not exist outside the context of an assembly.

52. How do you create a permanent cookie?

Setting the Expires property to MinValue means that the Cookie never expires.

53. What's a Windows process in .NET?

Windows process is an application that's running and had been allocated memory in .NET

54. What is Delegation in .NET?

A delegate acts like a strongly type function pointer. Delegates can invoke the methods that they reference without making explicit calls to those methods.
Delegate is an entity that is entrusted with the task of representation, assign or passing on information. In code sense, it means a Delegate is entrusted with a Method to report information back to it when a certain task (which the Method expects) is accomplished outside the Method's class.

55. What is Serialization in .NET?

The serialization is the process of converting the objects into stream of bytes.
they or used for transport the objects(via remoting) and persist objects(via files and databases)

56. Difference between Class And Interface in .NET?

Class is logical representation of object. It is collection of data and related sub procedures with definition.
Interface is also a class containing methods which is not having any definitions.
Class does not support multiple inheritance. But interface can support

57. Can any object be stored in a Viewstate in .NET?

An object that either is serializable or has a TypeConverter defined for it can be persisted in ViewState.

58 What is the use of ErrorProvider Control in .NET?

The ErrorProvider control is used to indicate invalid data on a data entry form. Using this control, you can attach error messages that display next to the control when the data is invalid, as seen in the following image. A red circle with an exclamation point blinks, and when the user mouses over the icon, the error message is displayed as a tooltip.

59. How do you validate the controls in an ASP .NET page?

Using special validation controls that are meant for validation of any controle.
We have Range Validator, Email Validator in .NET to validate any control.

60. How to manage pagination in a page using .NET?

Using pagination option in DataGrid control is available in .NET. We have to set the number of records for a page, then it takes care of pagination by itself automatically.

DBMS Interview Questions

1. What is database?

A database is a collection of information that is organized. So that it can easily be accessed, managed, and updated.

2. What is DBMS?

DBMS stands for Database Management System. It is a collection of programs that enables user to create and maintain a database.

3. What is a Database system?

The database and DBMS software together is called as Database system.

4. What are the advantages of DBMS?

I. Redundancy is controlled.

II. Providing multiple user interfaces.

III. Providing backup and recovery

IV. Unauthorized access is restricted.

V. Enforcing integrity constraints.

5. What is normalization?

It is a process of analysing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties
(1).Minimizing redundancy, (2). Minimizing insertion, deletion and update anomalies.

6. What is Data Model?
A collection of conceptual tools for describing data, data relationships data semantics and constraints.

7. What is E-R model?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

8. What is Object Oriented model?

This model is based on collection of objects. An object contains values stored in instance variables with in the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

9. What is an Entity?
An entity is a thing or object of importance about which data must be captured.

10. What is DDL (Data Definition Language)?

A data base schema is specifies by a set of definitions expressed by a special language called DDL.

11. What is DML (Data Manipulation Language)?

This language that enable user to access or manipulate data as organised by appropriate data model. Procedural DML or Low level: DML requires a user to specify what data are needed and how to get those data. Non-Procedural DML or High level: DML requires a user to specify what data are needed without specifying how to get those data

12. What is DML Compiler?

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand.

13. What is Query evaluation engine?

It executes low-level instruction generated by compiler.

14. What is Functional Dependency?

Functional Dependency is the starting point of normalization. Functional Dependency exists when a relation between two attributes allows you to uniquely determine the corresponding attribute's value.

15. What is 1 NF (Normal Form)?

The first normal form or 1NF is the first and the simplest type of normalization that can be implemented in a database. The main aims of 1NF are to:

1. Eliminate duplicative columns from the same table.

2. Create separate tables for each group of related data and identify each row with a unique column (the primary key).

16. What is Fully Functional dependency?

A functional dependency X Y is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

17. What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key.

18. What is 3NF?

A relation is in third normal form if it is in Second Normal Form and there are no functional (transitive) dependencies between two (or more) non-primary key attributes.

19. What is BCNF (Boyce-Codd Normal Form)?

A table is in Boyce-Codd normal form (BCNF) if and only if it is in 3NF and every determinant is a candidate key.

20. What is 4NF?

Fourth normal form requires that a table be BCNF and contain no multi-valued dependencies.

21. What is 5NF?

A table is in fifth normal form (5NF) or Project-Join Normal Form (PJNF) if it is in 4NF and it cannot have a lossless decomposition into any number of smaller tables.

22. What is a query?

A query with respect to DBMS relates to user commands that are used to interact with a data base.

23. What is meant by query optimization?

The phase that identifies an efficient execution plan for evaluating a query that has the least estimated cost is referred to as query optimization.

24. What is an attribute?
It is a particular property, which describes the entity.

25. What is RDBMS?

Relational Data Base Management Systems (RDBMS) are database management systems that maintain data records and indices in tables.

26. What's difference between DBMS and RDBMS?

DBMS provides a systematic and organized way of storing, managing and retrieving from collection of logically related information. RDBMS also provides what DBMS provides but above that it provides relationship integrity.

27. What is SQL?

SQL stands for Structured Query Language. SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database.

28. What is Stored Procedure?

A stored procedure is a named group of SQL statements that have been previously created and stored in the server database.

29. What is a view?

A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored.

30. What is Trigger?

A trigger is a SQL procedure that initiates an action when an event (INSERT, DELETE or UPDATE) occurs.

31. What is Index?

An index is a physical structure containing pointers to the data.

32. What is extension and intension?

Extension -It is the number of tuples present in a table at any instance. This is time dependent.

Intension -It is a constant value that gives the name, structure of table and the constraints laid on it.

33. What do you mean by atomicity and aggregation?

Atomicity-Atomicity states that database modifications must follow an “all or nothing” rule. Each transaction is said to be “atomic.” If one part of the transaction fails, the entire transaction fails.

Aggregation - A feature of the entity relationship model that allows a relationship set to participate in another relationship set. This is indicated on an ER diagram by drawing a dashed box around the aggregation.

34. What is RDBMS KERNEL?

Two important pieces of RDBMS architecture are the kernel, which is the software, and the data dictionary, which consists of the system- level data structures used by the kernel to manage the database.

35. Name the sub-systems of a RDBMS?

I/O, Security, Language Processing, Process Control, Storage Management, Logging and Recovery, Distribution Control, Transaction Control, Memory Management, Lock Management.

36. How do you communicate with an RDBMS?

You communicate with an RDBMS using Structured Query Language (SQL)

37. Disadvantage in File Processing System?

- Data redundancy & inconsistency.
- Difficult in accessing data.
- Data isolation.
- Data integrity.
- Concurrent access is not possible.
- Security Problems.

38. What is VDL (View Definition Language)?

It specifies user views and their mappings to the conceptual schema.

39. What is SDL (Storage Definition Language)?

This language is to specify the internal schema. This language may Specify the mapping between two schemas.

40. Describe concurrency control?

Concurrency control is the process managing simultaneous operations against a database so that database integrity is no compromised. There are two approaches to concurrency control.

The pessimistic approach involves locking and the optimistic approach involves versioning.

41. Describe the difference between homogeneous and heterogeneous distributed database?

A homogenous database is one that uses the same DBMS at each node. A heterogeneous database is one that may have a different DBMS at each node.

42. What is a distributed database?

A distributed database is a single logical database that is spread across more than one node or locations that are all connected via some communication link.

43. Explain the difference between two and three-tier architectures?

Three-tier architecture includes a client and two server layers.

The application code is stored on the application server and the database is stored on the database server. A two-tier architecture includes a client and one server layer. The database is stored on the database server.

44. Briefly describe the three types of SQL commands?

Data definition language commands are used to create, alter, and drop tables. Data manipulation commands are used to insert, modify, update, and query data in the database. Data control language commands help the DBA to control the database.

45. List some of the properties of a relation?

Relations in a database have a unique name and no multivalued attributes exist. Each row is unique and each attribute within a relation has a unique name. The sequence of both columns and rows is irrelevant.

46. Explain the differences between an intranet and an extranet?

An Internet database is accessible by everyone who has access to a Web site. An intranet database limits access to only people within a given organization.

47. What is SQL Deadlock?

Deadlock is a unique situation in a multi user system that causes two or more users to wait indefinitely for a locked resource.

48. What is a Catalog?

A catalog is a table that contains the information such as structure of each file, the type and storage format of each data item and various constraints on the data .The information stored in the catalog is called Metadata.

49. What is data ware housing & OLAP?

Data warehousing and OLAP (online analytical processing) systems are the techniques used in many companies to extract and analyze useful information from very large databases for decision making .

50. Describe the three levels of data abstraction?

Physical level: The lowest level of abstraction describes how data are stored.

Logical level: The next higher level of abstraction, describes what data are stored in database and what relationship among those data.

View level: The highest level of abstraction describes only part of entire database.

51. What is Data Independence?

Data independence means that the application is independent of the storage structure and access strategy of data.

52. How many types of relationship exist in database designing?

There are three major relationship models:-

One-to-one

One-to-many

Many-to-many

53. What is order by clause?

ORDER BY clause helps to sort the data in either ascending order to descending

54. What is the use of DBCC commands?

DBCC stands for database consistency checker. We use these commands to check the consistency of the databases, i.e., maintenance, validation task and status checks.

55. What is Collation?

Collation refers to a set of rules that determine how data is sorted and compared.

56. What is difference between DELETE & TRUNCATE commands?

Delete command removes the rows from a table based on the condition that we provide with a WHERE clause. Truncate will actually remove all the rows from a table and there will be no data in the table after we run the truncate command.

57. What is Hashing technique?

This is a primary file organization technique that provides very fast access to records on certain search conditions.

58. What is a transaction?

A transaction is a logical unit of database processing that includes one or more database access operations.

59. What are the different phases of Transaction?

Analysis phase

Redo phase

Undo phase

60. What is "transparent dbms"?

It is one, which keeps its physical structure hidden from user.

61. What are the primitive operations common to all record management System?

Addition, deletion and modification.

62. Explain the differences between structured data and unstructured data.

Structured data are facts concerning objects and events. The most important structured data are numeric, character, and dates.

Structured data are stored in tabular form. Unstructured data are multimedia data such as documents, photographs, maps, images, sound, and video clips. Unstructured data are most commonly found on Web servers and Web-enabled databases.

63. What are the major functions of the database administrator?

Managing database structure, controlling concurrent processing, managing processing rights and responsibilities, developing database security, providing for database recovery, managing the DBMS and maintaining the data repository.

64. What is a dependency graph?

A dependency graph is a diagram that is used to portray the connections between database elements.

65. Explain the difference between an exclusive lock and a shared lock?

An exclusive lock prohibits other users from reading the locked resource; a shared lock allows other users to read the locked resource, but they cannot update it.

66. Explain the "paradigm mismatch" between SQL and application programming languages.

SQL statements return a set of rows, while an application program works on one row at a time. To resolve this mismatch the results of SQL statements are processed as pseudofiles, using a cursor or pointer to specify which row is being processed.

67. Name four applications for triggers.

(1)Providing default values, (2) enforcing data constraints,

(3) Updating views and (4) enforcing referential integrity

68. What are the advantages of using stored procedures?

The advantages of stored procedures are (1) greater security, (2) decreased network traffic, (3) the fact that SQL can be optimized and (4) code sharing which leads to less work, standardized processing, and specialization among developers.

69. Explain the difference between attributes and identifiers.

Entities have attributes. Attributes are properties that describe the entity's characteristics. Entity instances have identifiers. Identifiers are attributes that name, or identify, entity instances.

70. What is Enterprise Resource Planning (ERP), and what kind of a database is used in an ERP application?

Enterprise Resource Planning (ERP) is an information system used in manufacturing companies and includes sales, inventory, production planning, purchasing and other business functions. An ERP system typically uses a multiuser database.

71. Describe the difference between embedded and dynamic SQL?

Embedded SQL is the process of including hard coded SQL statements. These statements do not change unless the source code is modified. Dynamic SQL is the process of generating SQL on the fly.The statements generated do not have to be the same each time.

72. Explain a join between tables

A join allows tables to be linked to other tables when a relationship between the tables exists. The relationships are established by using a common column in the tables and often uses the primary/foreign key relationship.

73. Describe a subquery.

A subquery is a query that is composed of two queries. The first query (inner query) is within the WHERE clause of the other query (outer query).

74. Compare a hierarchical and network database model?

The hierarchical model is a top-down structure where each parent may have many children but each child can have only one parent. This model supports one-to-one and one-to-many relationships.

The network model can be much more flexible than the hierarchical model since each parent can have multiple children but each child can also have multiple parents. This model supports one-to-one, one-to-many, and many-to-many relationships.

75. Explain the difference between a dynamic and materialized view.

A dynamic view may be created every time that a specific view is requested by a user. A materialized view is created and or updated infrequently and it must be synchronized with its associated base table(s).

76. Explain what needs to happen to convert a relation to third normal form.

First you must verify that a relation is in both first normal form and second normal form. If the relation is not, you must convert into second normal form. After a relation is in second normal form, you must remove all transitive dependencies.

77. Describe the four types of indexes?

A unique primary index is unique and is used to find and store a row. A nonunique primary index is not unique and is used to find a row but also where to store a row (based on its unique primary index). A unique secondary index is unique for each row and used to find table rows. A nonunique secondary index is not unique and used to find table rows.

78. Explain minimum and maximum cardinality?

Minimum cardinality is the minimum number of instances of an entity that can be associated with each instance of another entity. Maximum cardinality is the maximum number of instances of an entity that can be associated with each instance of another entity.

79. What is deadlock? How can it be avoided? How can it be resolved once it occurs?

Deadlock occurs when two transactions are each waiting on a resource that the other transaction holds. Deadlock can be prevented by requiring transactions to acquire all locks at the same time; once it occurs, the only way to cure it is to abort one of the transactions and back out of partially completed work.

80. Explain what we mean by an ACID transaction.

An ACID transaction is one that is atomic, consistent, isolated, and durable. Durable means that database changes are permanent. Consistency can mean either statement level or transaction level consistency. With transaction level consistency, a transaction may not see its own changes.Atomic means it is performed as a unit.

81. Under what conditions should indexes be used?

Indexes can be created to enforce uniqueness, to facilitate sorting, and to enable fast retrieval by column values. A good candidate for an index is a column that is frequently used with equal conditions in WHERE clauses.

82. What is difference between SQL and SQL SERVER?

SQL is a language that provides an interface to RDBMS, developed by IBM. SQL SERVER is a RDBMS just like Oracle, DB2.

83. What is Specialization?

It is the process of defining a set of subclasses of an entity type where each subclass contain all the attributes and relationships of the parent entity and may have additional attributes and relationships which are specific to itself.

84. What is generalization?

It is the process of finding common attributes and relations of a number of entities and defining a common super class for them.

85. What is meant by Proactive, Retroactive and Simultaneous Update?

Proactive Update: The updates that are applied to database before it becomes effective in real world.

Retroactive Update: The updates that are applied to database after it becomes effective in real world.

Simultaneous Update: The updates that are applied to database at the same time when it becomes effective in real world.

86. What is RAID Technology?

Redundant array of inexpensive (or independent) disks. The main goal of raid technology is to even out the widely different rates of performance improvement of disks against those in memory and microprocessor. Raid technology employs the technique of data striping to achieve higher transfer rates.

87. What are serial, non serial schedule?

A schedule S is serial if, for every transaction T participating in the schedule, all the operations of T is executed consecutively in the schedule, otherwise, the schedule is called non-serial schedule.

88. What are conflict serializable schedules?

A schedule S of n transactions is serializable if it is equivalent to some serial schedule of the same n transactions.

89. What is view serializable?

A schedule is said to be view serializable if it is view equivalent with some serial schedule.

90. What is a foreign key?

A key of a relation schema is called as a foreign key if it is the primary key of some other relation to which it is related to.

91. What are the disadvantages of using a dbms?

1) High initial investments in h/w, s/w, and training.
2) Generality that a DBMS provides for defining and processing data.
3) Overhead for providing security, concurrency control, recovery, and integrity functions.

92. What is Lossless join property?
It guarantees that the spurious tuple generation does not occur with respect to relation schemas after decomposition.

93. What is a Phantom Deadlock?
In distributed deadlock detection, the delay in propagating local information might cause the deadlock detection algorithms to identify deadlocks that do not really exist. Such situations are called phantom deadlocks and they lead to unnecessary aborts.

94. What is a checkpoint and When does it occur?
A Checkpoint is like a snapshot of the DBMS state. By taking checkpoints, the DBMS can reduce the amount of work to be done during restart in the event of subsequent crashes.

95. What is schema?
The description of a data base is called the database schema , which is specified during

database design and is not expected to change frequently . A displayed schema is called schema diagram .We call each object in the schema as schema construct.

Data Structure Interview Questions

1) What is data structure?
Data structure refers to the way data is organized and manipulated. It seeks to find ways to make data access more efficient. When dealing with data structure, we not only focus on one piece of data, but rather different set of data and how they can relate to one another in an organized manner.

2) Differentiate file structure from storage structure.
Basically, the key difference is the memory area that is being accessed. When dealing with the
structure that resides the main memory of the computer system, this is referred to as storage structure. When dealing with an auxiliary structure, we refer to it as file structure.

3) When is a binary search best applied?
A binary search is an algorithm that is best applied to search a list when the elements are already in order or sorted. The list is search starting in the middle, such that if that middle value
is not the target search key, it will check to see if it will continue the search on the lower half of
the list or the higher half. The split and search will then continue in the same manner.

4) What is a linked list?
A linked list is a sequence of nodes in which each node is connected to the node following it. This forms a chain-like link of data storage.

5) How do you reference all the elements in a one-dimension array?
To do this, an indexed loop is used, such that the counter runs from 0 to the array size minus one. In this manner, we are able to reference all the elements in sequence by using the loop counter as the array subscript.

6) In what areas do data structures applied?
Data structure is important in almost every aspect where data is involved. In general, algorithms
that involve efficient data structure is applied in the following areas: numerical analysis, operating system, A.I., compiler design, database management, graphics, and statistical analysis, to name a few.

7) What is LIFO?
LIFO is short for Last In First Out, and refers to how data is accessed, stored and retrieved. Using this scheme, data that was stored last , should be the one to be extracted first. This also

means that in order to gain access to the first data, all the other data that was stored before this first data must first be retrieved and extracted.

8) What is a queue?
A queue is a data structure that can simulates a list or stream of data. In this structure, new elements are inserted at one end and existing elements are removed from the other end.

9) What are binary trees?
A binary tree is one type of data structure that has two nodes, a left node and a right node. In programming, binary trees are actually an extension of the linked list structures.

10) Which data structure is applied when dealing with a recursive function?
Recursion, which is basically a function that calls itself based on a terminating condition, makes use of the stack. Using LIFO, a call to a recursive function saves the return address so that it knows how to return to the calling function after the call terminates.

11) What is a stack?
A stack is a data structure in which only the top element can be accessed. As data is stored in the stack, each data is pushed downward, leaving the most recently added data on top.

12) Explain Binary Search Tree
A binary search tree stores data in such a way that they can be retrieved very efficiently. The left subtree contains nodes whose keys are less than the node's key value, while the right subtree contains nodes whose keys are greater than or equal to the node's key value. Moreover, both subtrees are also binary search trees.

13) What are multidimensional arrays?
Multidimensional arrays make use of multiple indexes to store data. It is useful when storing data that cannot be represented using a single dimensional indexing, such as data representation in a board game, tables with data stored in more than one column.

14) Are linked lists considered linear or non-linear data structure?
It actually depends on where you intend to apply linked lists. If you based it on storage, a linked list is considered non-linear. On the other hand, if you based it on access strategies, then a linked list is considered linear.

15) How does dynamic memory allocation help in managing data?
Aside from being able to store simple structured data types, dynamic memory allocation can combine separately allocated structured blocks to form composite structures that expand and contract as needed.

16) What is FIFO?
FIFO is short for First-in, First-out, and is used to represent how data is accessed in a queue. Data has been inserted into the queue list the longest is the one that is removed first.

17) What is an ordered list?
An ordered list is a list in which each node's position in the list is determined by the value of its key component, so that the key values form an increasing sequence, as the list is traversed.

18) What is merge sort?
Merge sort takes a divide-and-conquer approach to sorting data. In a sequence of data, adjacent ones are merged and sorted to create bigger sorted lists. These sorted lists are then merged again to form an even bigger sorted list, which continuous until you have one single sorted list.

19) Differentiate NULL and VOID.
Null is actually a value, whereas Void is a data type identifier. A variable that is given a Null value simply indicates an empty value. Void is used to identify pointers as having no initial size.

20) What is the primary advantage of a linked list?
A linked list is a very ideal data structure because it can be modified easily. This means that modifying a linked list works regardless of how many elements are in the list.

21) What is the difference between a PUSH and a POP?
Pushing and popping applies to the way data is stored and retrieved in a stack. A push denotes
data being added to it, meaning data is being “pushed” into the stack. On the other hand, a pop
denotes data retrieval, and in particular refers to the topmost data being accessed.

22) What is a linear search?
A linear search refers to the way a target key is being searched in a sequential data structure. Using this method, each element in the list is checked and compared against the target key, and is repeated until found or if the end of the list has been reached.

23) How does variable declaration affect memory allocation?
The amount of memory to be allocated or reserved would depend on the data type of the variable being declared. For example, if a variable is declared to be of integer type, then 32 bits
of memory storage will be reserved for that variable.

24) What is the advantage of the heap over a stack?
Basically, the heap is more flexible than the stack. That's because memory space for the heap can be dynamically allocated and de-allocated as needed. However, memory of the heap can at
times be slower when compared to that stack.

25) What is a postfix expression?
A postfix expression is an expression in which each operator follows its operands. The advantage of this form is that there is no need to group sub-expressions in parentheses or to consider operator precedence.

26) What is Data abstraction?
Data abstraction is a powerful tool for breaking down complex data problems into manageable chunks. This is applied by initially specifying the data objects involved and the operations to be performed on these data objects without being overly concerned with how the data objects will be represented and stored in memory.

27) How do you insert a new item in a binary search tree?
Assuming that the data to be inserted is a unique value (that is, not an existing entry in the tree), check first if the tree is empty. If it's empty, just insert the new item in the root node. If it's not empty, refer to the new item's key. If it's smaller than the root's key, insert it into the root's left subtree, otherwise, insert it into the root's right subtree.

28) How does a selection sort work for an array?
The selection sort is a fairly intuitive sorting algorithm,, though not necessarily efficient. To perform this, the smallest element is first located and switched with the element at subscript zero, thereby placing the smallest element in the first position. The smallest element remaining in the subarray is then located next with subscripts 1 through n-1 and switched with the element
at subscript 1, thereby placing the second smallest element in the second position. The steps are repeated in the same manner till the last element.

29) How do signed and unsigned numbers affect memory?
In the case of signed numbers, the first bit is used to indicate whether positive or negative,

which leaves you with one bit short. With unsigned numbers, you have all bits available for that number. The effect is best seen in the number range (unsigned 8 bit number has a range 0-255, while 8-bit signed number has a range -128 to +127.

30) What is the minimum number of nodes that a binary tree can have?
A binary tree can have a minimum of zero nodes, which occurs when the nodes have NULL values. Furthermore, a binary tree can also have 1 or 2 nodes.

31) What are dynamic data structures?
Dynamic data structures are structures that expand and contract as a program runs. It provides a flexible means of manipulating data because it can adjust according to the size of the data.

32) In what data structures are pointers applied?
Pointers that are used in linked list have various applications in data structure. Data structures that make use of this concept include the Stack, Queue, Linked List and Binary Tree.

33) Do all declaration statements result in a fixed reservation in memory?
Most declarations do, with the exemption of pointers. Pointer declaration does not allocate memory for data, but for the address of the pointer variable. Actual memory allocation for the data comes during run-time.

34) What are ARRAYs?
When dealing with arrays, data is stored and retrieved using an index that actually refers to the element number in the data sequence. This means that data can be accessed in any order. In programming, an array is declared as a variable having a number of indexed elements.

35) What is the minimum number of queues needed when implementing a priority queue?
The minimum number of queues needed in this case is two. One queue is intended for sorting priorities while the other queue is intended for actual storage of data.

36) Which sorting algorithm is considered the fastest?
There are many types of sorting algorithms: quick sort, bubble sort, balloon sort, radix sort, merge sort, etc. Not one can be considered the fastest because each algorithm is designed for a particular data structure and data set. It would depend on the data set that you would want to sort.

37) Differentiate STACK from ARRAY.
Data that is stored in a stack follows a LIFO pattern. This means that data access follows a sequence wherein the last data to be stored will the first one to be extracted. Arrays, on the other hand, does not follow a particular order and instead can be accessed by referring to the indexed element within the array.

38) Give a basic algorithm for searching a binary search tree.
1. if the tree is empty, then the target is not in the tree, end search
2. if the tree is not empty, the target is in the tree
3. check if the target is in the root item
4. if target is not in the root item, check if target is smaller than the root's value
5. if target is smaller than the root's value, search the left subtree
6. else, search the right subtree

39) What is a dequeue?
A dequeue is a double-ended queue. This is a structure wherein elements can be inserted or removed from either end.

40) What is a bubble sort and how do you perform it?
A bubble sort is one sorting technique that can be applied to data structures such as an array. It works by comparing adjacent elements and exchanges their values if they are out of order. This method lets the smaller values “bubble” to the top of the list, while the larger value sinks to the bottom.

41) What are the parts of a linked list?
A linked list typically has two parts: the head and the tail. Between the head and tail lie the actual nodes, with each node being linked in a sequential manner.

42) How does selection sort work?
Selection sort works by picking the smallest number from the list and placing it at the front. This
process is repeated for the second position towards the end of the list. It is the simplest sort algorithm.

43) What is a graph?
A graph is one type of data structure that contains a set of ordered pairs. These ordered pairs are also referred to as edges or arcs, and are used to connect nodes where data can be stored

and retrieved.

44) Differentiate linear from non linear data structure.
Linear data structure is a structure wherein data elements are adjacent to each other. Examples
of linear data structure include arrays, linked lists, stacks and queues. On the other hand, nonlinear
data structure is a structure wherein each data element can connect to more than two adjacent data elements. Examples of non linear data structure include trees and graphs.

45) What is an AVL tree?
An AVL tree is a type of binary search tree that is always in a state of partially balanced. The balance is measured as a difference between the heights of the subtrees from the root. This selfbalancing tree was known to be the first data structure to be designed as such.

46) What are doubly linked lists?
Doubly linked lists are a special type of linked list wherein traversal across the data elements can be done in both directions. This is made possible by having two links in every node, one that links to the next node and other one that links to the previous node.

47) What is Huffman's algorithm?
Huffman's algorithm is associated in creating extended binary trees that has minimum weighted
path lengths from the given weights. It makes use of a table that contains frequency of occurrence for each data element.

48) What is Fibonacci search?
Fibonacci search is a search algorithm that applies to a sorted array. It makes use of a divideand-
conquer approach that can greatly reduce the time needed in order to reach the target element.

49) Briefly explain recursive algorithm.
Recursive algorithm targets a problem by dividing it into smaller, manageable sub-problems. The output of one recursion after processing one sub-problem becomes the input to the next recursive process.

50) How do you search for a target key in a linked list?
To find the target key in a linked list, you have to apply sequential search. Each node is traversed and compared with the target key, and if it is different, then it follows the link to the next node. This traversal continues until either the target key is found or if the last node is reached.

Operating System Interview Questions

1. What is an operating system?

An operating system is a program that acts as an intermediary between the user and the computer hardware. The purpose of an OS is to provide a convenient environment in which user can execute programs in a convenient and efficient manner.

2. What are the different operating systems?

1. Batched operating systems
2. Multi-programmed operating systems
3. timesharing operating systems
4. Distributed operating systems
5. Real-time operating systems

3. What are the basic functions of an operating system?

Operating system controls and coordinates the use of the hardware among the various applications programs for various uses. Operating system acts as resource allocator and manager. Also operating system is control program which controls the user programs to prevent errors and improper use of the computer. It is especially concerned with the operation and control of I/O devices.

4. What is kernel?

Kernel is the core and essential part of computer operating system that provides basic services for all parts of OS.

5. What is difference between micro kernel and macro kernel?

Micro kernel is a kernel which run services those are minimal for operating system performance. In this kernel all other operations are performed by processor.

Macro Kernel is a combination of micro and monolithic kernel. In monolithic kernel all operating system code is in single executable image.

6. What is dead lock?

Deadlock is a situation or condition where the two processes are waiting for each other to complete so that they can start. This result both the processes to hang.

7. What is a process?

A program in execution is called a process.

Processes are of two types:
1. Operating system processes
2. User processes

8. What are the states of a process?

1. New
2. Running
3. Waiting
4. Ready
5. Terminated

9. What is starvation and aging?

Starvation is Resource management problem where a process does not get the resources it needs for a long time because the resources are being allocated to other processes.

Aging is a technique to avoid starvation in a scheduling system.

10. What is semaphore?

Semaphore is a variable, whose status reports common resource, Semaphore is of two types one is Binary semaphore and other is Counting semaphore.

11. What is context switching?

Transferring the control from one process to other process requires saving the state of the old process and loading the saved state for new process. This task is known as context switching.

12. What is a thread?

A thread is a program line under execution. Thread sometimes called a light-weight process, is a basic unit of CPU utilization; it comprises a thread id, a program counter, a register set, and a stack

13. What is process synchronization?

A situation, where several processes access and manipulate the same data concurrently and the outcome of the execution depends on the particular order in which the access takes place, is called race condition. To guard against the race condition we need to ensure that only one process at a time can be manipulating the same data. The technique we use for this is called process synchronization.

14. What is virtual memory?

Virtual memory is hardware technique where the system appears to have more memory that it actually does. This is done by time-sharing, the physical memory and storage parts of the memory one disk when they are not actively being used.

15. What is thrashing?

It is a phenomenon in virtual memory schemes when the processor spends most of its time swapping pages, rather than executing instructions. This is due to an inordinate number of page faults.

16. What is fragmentation? Tell about different types of fragmentation?

When many of free blocks are too small to satisfy any request then fragmentation occurs. External fragmentation and internal fragmentation are two types of fragmentation. External Fragmentation happens when a dynamic memory allocation algorithm allocates some memory and a small piece is left over that cannot be effectively used. Internal fragmentation is the space wasted inside of allocated memory blocks because of restriction on the allowed sizes of allocated blocks.

17. What are necessary conditions for dead lock?

1. Mutual exclusion (where at least one resource is non-sharable)

2. Hold and wait (where a process holds one resource and waits for other resource)

3. No preemption (where the resources can’t be preempted)

4. Circular wait (where p[i] is waiting for p[j] to release a resource. i= 1,2,…n

j=if (i!=n) then i+1

else 1)

18. What is cache memory?

Cache memory is random access memory (RAM) that a computer microprocessor can access more quickly than it can access regular RAM. As the microprocessor processes data, it looks first in the cache memory and if it finds the data there (from a previous reading of data), it does not have to do the more time-consuming reading of data from larger memory.

19. What is logical and physical addresses space?

Logical address space is generated from CPU; it bound to a separate physical address space is central to proper memory management. Physical address space is seen by the memory unit. Logical address space is virtual address space. Both these address space will be same at compile time but differ at execution time.

20. Differentiate between Complier and Interpreter?

An interpreter reads one instruction at a time and carries out the actions implied by that instruction. It does not perform any translation. But a compiler translates the entire instructions

21. What is Throughput, Turnaround time, waiting time and Response time?

Throughput – number of processes that complete their execution per time unit

Turnaround time – amount of time to execute a particular process

Waiting time – amount of time a process has been waiting in the ready queue

Response time – amount of time it takes from when a request was submitted until the first response is produced, not output (for time-sharing environment)

22. What is Memory-Management Unit (MMU)?

Hardware device that maps virtual to physical address. In MMU scheme, the value in the relocation register is added to every address generated by a user process at the time it is sent to memory.

->The user program deals with logical addresses; it never sees the real physical addresses

23. What is a Real-Time System?

A real time process is a process that must respond to the events within a certain time period. A real time operating system is an operating system that can run real time processes successfully

24. What is a trap and trapdoor?

Trapdoor is a secret undocumented entry point into a program used to grant access without normal methods of access authentication. A trap is a software interrupt, usually the result of an error condition.

25. When is a system in safe state?

The set of dispatchable processes is in a safe state if there exists at least one temporal order in which all processes can be run to completion without resulting in a deadlock.

26. Explain the concept of the Distributed systems?

Distributed systems work in a network. They can share the network resources, communicate with each other.

27. What is cache-coherency?

In a multiprocessor system there exist several caches each may containing a copy of same variable A. Then a change in one cache should immediately be reflected in all other caches this process of maintaining the same value of a data in all the caches s called cache-coherency.

28. What is a long term scheduler & short term schedulers?

Long term schedulers are the job schedulers that select processes from the job queue and load them into memory for execution. The short term schedulers are the CPU schedulers that select a process from the ready queue and allocate the CPU to one of them.

29. Explain the meaning of mutex.

Mutex is the short form for 'Mutual Exclusion object'. A mutex allows multiple threads for sharing the same resource. The resource can be file. A mutex with a unique name is created at the time of starting a program. A mutex must be locked from other threads, when any thread that needs the resource. When the data is no longer used / needed, the mutex is set to unlock.

30. What is cycle stealing?

We encounter cycle stealing in the context of Direct Memory Access (DMA). Either the DMA controller can use the data bus when the CPU does not need it, or it may force the CPU to temporarily suspend operation. The latter technique is called cycle stealing. Note that cycle stealing can be done only at specific break points in an instruction cycle.

31. What is Marshalling?

The process of packaging and sending interface method parameters across thread or process boundaries.

32. What is a daemon?

Daemon is a program that runs in the background without user's interaction. A daemon runs in a multitasking operating system like UNIX. A daemon is initiated and controlled by special programs known as 'processes'.

33. What is pre-emptive and non-preemptive scheduling?

Preemptive scheduling: The preemptive scheduling is prioritized. The highest priority process should always be the process that is currently utilized.

Non-Preemptive scheduling: When a process enters the state of running, the state of that process is not deleted from the scheduler until it finishes its service time.

34. What is busy waiting?

The repeated execution of a loop of code while waiting for an event to occur is called busy-waiting. The CPU is not engaged in any real productive activity during this period, and the process does not progress toward completion.

35. What is page cannibalizing?

Page swapping or page replacements are called page cannibalizing.

36. What is SMP?

To achieve maximum efficiency and reliability a mode of operation known as symmetric multiprocessing is used. In essence, with SMP any process or threads can be assigned to any processor.

37. What is process migration?

It is the transfer of sufficient amount of the state of process from one machine to the target machine.

38. Difference between Primary storage and secondary storage?

Primary memory is the main memory (Hard disk, RAM) where the operating system resides.

Secondary memory can be external devices like CD, floppy magnetic discs etc. secondary storage cannot be directly accessed by the CPU and is also external memory storage.

39. Define compactions.

Compaction is a process in which the free space is collected in a large memory chunk to make some space available for processes.

40. What are residence monitors?

Early operating systems were called residence monitors.

41. What is dual-mode operation?

In order to protect the operating systems and the system programs from the malfunctioning programs the two mode operations were evolved
System mode
User mode.

42. What is a device queue?

A list of processes waiting for a particular I/O device is called device queue.

43. What are the different types of Real-Time Scheduling?

Hard real-time systems required to complete a critical task within a guaranteed amount of time.
Soft real-time computing requires that critical processes receive priority over less fortunate ones.

44. What is relative path and absolute path?

Absolute path-- Exact path from root directory.
Relative path-- Relative to the current path.

45. What are the disadvantages of context switching?

Time taken for switching from one process to other is pure over head. Because the system does no useful work while switching. So one of the solutions is to go for threading when ever possible.

46. What is a data register and address register?

Data registers - can be assigned to a variety of functions by the programmer. They can be used with any machine instruction that performs operations on data.
Address registers - contain main memory addresses of data and instructions or they contain a portion of the address that is used in the calculation of the complete addresses.

47. What is DRAM?

Dynamic Ram stores the data in the form of Capacitance, and Static RAM stores the data in Voltages.

48. What are local and global page replacements?

Local replacement means that an incoming page is brought in only to the relevant process' address space. Global replacement policy allows any page frame from any process to be replaced. The latter is applicable to variable partitions model only.

49. Explain the concept of the batched operating systems?

In batched operating system the users gives their jobs to the operator who sorts the programs according to their requirements and executes them. This is time consuming but makes the CPU busy all the time.

50. What is SCSI?

SCSI - Small computer systems interface is a type of interface used for computer components such as hard drives, optical drives, scanners and tape drives. It is a competing technology to standard IDE (Integrated Drive Electronics).

51.When is a system in safe state?

The set of dispatchable processes is in a safe state if there exists at least one temporal order in which all processes can be run to completion without resulting in a deadlock.

52. What is cycle stealing?

We encounter cycle stealing in the context of Direct Memory Access (DMA). Either the DMA controller can use the data bus when the CPU does not need it, or it may force the CPU to temporarily suspend operation. The latter technique is called cycle stealing. Note that cycle stealing can be done only at specific break points in an instruction cycle.

53. What is an idle thread?

The special thread a dispatcher will execute when no ready thread is found.

54. What is FtDisk?

It is a fault tolerance disk driver for Windows NT.

55.What is Dispatcher?

Dispatcher module gives control of the CPU to the process selected by the short-term scheduler; this involves: Switching context, Switching to user mode, Jumping to the proper location in the user program to restart that program, dispatch latency – time it takes for the dispatcher to stop one process and start another running.

56. When does the condition 'rendezvous' arise?

In message passing, it is the condition in which, both, the sender and receiver are blocked until the message is delivered.

57. What is process spawning?

When the OS at the explicit request of another process creates a process, this action is called process spawning

58. What are the reasons for process suspension?
1) swapping
2) interactive user request
3) timing
4) parent process request

59. What are the sub-components of I/O manager in Windows NT?
1) Network redirector/ Server
2) Cache manager.
3) File systems
4) Network driver
5) Device driver

60. What is a drawback of MVT?
1) ability to support multiple processors
2) virtual storage
3) source level debugging

Computer Networking Interview Questions

1. Define Network?
A network is a set of devices connected by physical media links. A network is recursively is a connection of two or more nodes by a physical link or two or more networks connected by one or more nodes.

2. What is Protocol?
A protocol is a set of rules that govern all aspects of information communication.

3. What is a Link?
At the lowest level, a network can consist of two or more computers directly connected by some physical medium such as coaxial cable or optical fiber. Such a physical medium is called as Link.

4. What is a node?
A network can consist of two or more computers directly connected by some physical medium such as coaxial cable or optical fiber. Such a physical medium is called as Links and the computer it connects is called as Nodes.

5. What is a gateway or Router?
A node that is connected to two or more networks is commonly called as router or Gateway. It generally forwards message from one network to another.

6. Name the factors that affect the performance of the network?
a.Number of Users
b. Type of transmission medium
c. Hardware
d. Software

7. What is Round Trip Time?
The duration of time it takes to send a message from one end of a network to the other and back, is called RTT.

8. List the layers of OSI
a. Physical Layer
b. Data Link Layer
c. Network Layer
d. Transport Layer
e. Session Layer
f. Presentation Layer
g. Application Layer

9. Which layers are network support layers?
a. Physical Layer
b. Data link Layer and
c. Network Layers

10. Which layers are user support layers?
a. Session Layer
b. Presentation Layer and
c. Application Layer

11. What is Pipelining ?
In networking and in other areas, a task is often begun before the previous task has ended. This is known as pipelining.

12. What is Piggy Backing?
A technique called piggybacking is used to improve the efficiency of the bidirectional protocols. When a frame is carrying data from A to B, it can also carry control information about arrived (or lost) frames from B; when a frame is carrying data from B to A, it can also carry control information about the arrived (or lost) frames from A.

13. What are the two types of transmission technology available?
(i) Broadcast and (ii) point-to-point

14. What is Bandwidth?
Every line has an upper limit and a lower limit on the frequency of signals it can carry. This limited range is called the bandwidth.

15. Explain RIP (Routing Information Protocol)
It is a simple protocol used to exchange information between the routers.

16. What is subnet?
A generic term for section of a large networks usually separated by a bridge or router.

17. What is MAC address?
The address for a device as it is identified at the Media Access Control (MAC) layer in the network architecture. MAC address is usually stored in ROM on the network adapter card and is unique.

18. What is multiplexing?
Multiplexing is the process of dividing a link, the phycal medium, into logical channels for better efficiency. Here medium is not changed but it has several channels instead of one.

19. What is simplex?
It is the mode of communication between two devices in which flow of data is unidirectional. i.e. one can transmit and other can receive.
E.g. keyboard and monitor.

20. What is half-duplex?
It is the mode of communication between two devices in which flow of data is bi-directional but not at the same time. ie each station can transmit and receive but not at the same time.
E.g walkie-talkies are half-duplex system.

21.What is full duplex?
It is the mode of communication between two devices in which flow of data is bi-directional and it occurs simultaneously. Here signals going in either direction share the capacity of the link.
E.g. telephone

22. What is sampling?
It is the process of obtaining amplitude of a signal at regular intervals.

23. What is Asynchronous mode of data transmission?
It is a serial mode of transmission.
In this mode of transmission, each byte is framed with a start bit and a stop bit. There may be a variable length gap between each byte.

24. What is Synchronous mode of data transmission?
It is a serial mode of transmission.In this mode of transmission, bits are sent in a continuous stream without start and stop bit and without gaps between bytes. Regrouping the bits into meaningful bytes is the responsibility of the receiver.

25. What are the different types of multiplexing?
Multiplexing is of three types. Frequency division multiplexing and wave division multiplexing is for analog signals and time division multiplexing is for digital signals.

26. What are the different transmission media?
The transmission media is broadly categorized into two types
i)Guided media(wired)
i)Unguided media(wireless)

27. What are the duties of data link layer?
Data link layer is responsible for carrying packets from one hop (computer or router) to the next. The duties of data link layer include packetizing, adderssing, error control, flow control, medium access control.

28. .What are the types of errors?
Errors can be categorized as a single-bit error or burst error. A single bit error has one bit error per data unit. A burst error has two or more bits errors per data unit.

29. What do you mean by redundancy?
Redundancy is the concept of sending extra bits for use in error detection. Three common redundancy methods are parity check, cyclic redundancy check (CRC), and checksum.

30. Define parity check.
In parity check, a parity bit is added to every data unit so that the total number of 1s is even

(or odd for odd parity).Simple parity check can detect all single bit errors. It can detect burst errors only if the total number of errors in each data unit is odd.In two dimensional parity checks, a block of bits is divided into rows and a redundant row of bits is added to the whole block.

31. Define cyclic redundancy check (CRC).
C RC appends a sequence of redundant bits derived from binary division to the data unit. The divisor in the CRC generator is often represented as an algebraic polynomial.

32. What is hamming code?
The hamming code is an error correction method using redundant bits. The number of bits is a function of the length of the data bits. In hamming code for a data unit of m bits, we use the formula 2r >= m+r+1 to determine the number of redundant bits needed. By rearranging the order of bit transmission of the data units, the hamming code can correct burst errors.

33.Define stop and wait ARQ.
In stop and wait ARQ, the sender sends a frame and waits for an acknowledgement from the receiver before sending the next frame.

34. What do you mean by network control protocol?
Network control protocol is a set of protocols to allow the encapsulation of data coming from network layer protocol that requires the services of PPP

35. What do you mean by CSMA?
To reduce the possibility of collision CSMA method was developed. In CSMA each station first listen to the medium (Or check the state of the medium) before sending. It can't eliminate collision.

36. What do you mean by Bluetooth?
It is a wireless LAN technology designed to connect devices of different functions such as telephones, notebooks, computers, cameras, printers and so on.

37. What is IP address?
The internet address (IP address) is 32bits that uniquely and universally defines a host or router on the internet.The portion of the IP address that identifies the network is called netid. The portion of the IP address that identifies the host or router on the network is called hostid.

38. What do you mean by ALOHA ?
It is the method used to solve the channel allocation problem .It is used for:
i)ground based radio broadcasting
ii)In a network in which uncoordinated users are competing for the use of single channel.
It is of two types:
1.Pure aloha
2.Slotted aloha

39. What is Firewalls?
It is an electronic downbridge which is used to enhance the security of a network. It's configuration has two components.
i)Two routers
ii)Application gateway
the packets traveling through the LAN are inspected here and packets meeting certain criteria are forwarded and others are dropped.

40. What is Repeaters ?
A receiver receives a signal before it becomes too weak or corrupted,regenerates the original bit pattern,and puts the refreshed copy back onto the link.It operates on phycal layer of OSI model.

41. What is Bridges?
They divide large network into smaller components.They can relay frames between two originally separated LANs.They provide security through partitioning traffic.They operate on physical and data link layer of OSI model.

42. What is ICMP?
ICMP is Internet Control Message Protocol, a network layer protocol of the TCP/IP suite used by hosts and gateways to send notification of datagram problems back to the sender. It uses the echo test / reply to test whether a destination is reachable and responding. It also handles both control and error messages.
.

43. What is FDM?
FDM is an analog technique that can be applied when the bandwidth of a link is greater than the combined bandwidths of the signals to be transmitted.

44. What is WDM?
WDM is conceptually the same as FDM, except that the multiplexing and demultiplexing involve light signals transmitted through fiber optics channel.

45. What is TDM?
TDM is a digital process that can be applied when the data rate capacity of the transmission medium is greater than the data rate required by the sending and receiving devices.

46. List the steps involved in creating the checksum.
a. Divide the data into sections
b. Add the sections together using 1's complement arithmetic
c. Take the complement of the final sum, this is the checksum.

47. Compare Error Detection and Error Correction:
The correction of errors is more difficult than the detection. In error detection, checks only any error has occurred. In error correction, the exact number of bits that are corrupted and location in the message are known. The number of the errors and the size of the message are important factors.

48. What are the protocols in application layer ?
The protocols defined in application layer are
• TELNET
• FTP
• SMTP
• DNS

49. What are the protocols in transport layer ?
The protocols defined in transport layer are
• TCP
• UDP

50. What do you mean by client server model ?
In client server model ,the client runs a program to request a service and the server runs a program to provide the service.These two programs communicate with each other. One server program can provide services to many client programs.

51. What is TELNET ?
TELNET is a client –server application that allows a user to log on to a remote machine,giving the user access to the remote system. TELNET is an abbreviation of terminal Network.

52. What is Hypertext Transfer Protocol(HTTP) ?
It is the main protocol used to access data on the World Wide Web .the protol transfers data in the form of plain text,hypertext,audio,video,and so on. It is so called because its efficiency allows its use in a hypertext environment where there are rapid jumps from one document to another.

53. What is World Wide Web ?
Ans: World Wide Web is a repository of information spread all over the world and linked together.It is a unique combination of flexibility,portability,and user-friendly features .The World Wide Web today is a distributed client-server service,in which a client using a browser can access a service using a server.The service provided is distributed over many locations called web sites.

54. What is Beaconing?
The process that allows a network to self-repair networks problems. The stations on the network notify the other stations on the ring when they are not receiving the transmissions. Beaconing is used in Token ring and FDDI networks.

55. What is RAID?
A method for providing fault tolerance by using multiple hard disk drives.

56. What is NETBIOS and NETBEUI?
NETBIOS is a programming interface that allows I/O requests to be sent to and received from a remote computer and it hides the networking hardware from applications.
NETBEUI is NetBIOS extended user interface. A transport protocol designed by microsoft and IBM for the use on small subnets.

57. What is difference between ARP and RARP?
The address resolution protocol (ARP) is used to associate the 32 bit IP address with the 48 bit physical address, used by a host or a router to find the physical address of another host on its network by sending a ARP query packet that includes the IP address of the receiver.
The reverse address resolution protocol (RARP) allows a host to discover its Internet address when it knows only its physical address.

58. What is the minimum and maximum length of the header in the TCP segment and IP datagram?
The header should have a minimum length of 20 bytes and can have a maximum length of 60 bytes.

59. What are major types of networks and explain?
Server-based network: provide centralized control of network resources and rely on server computers to provide security and network administration
Peer-to-peer network: computers can act as both servers sharing resources and as clients using the resources.

60. What are the important topologies for networks?
BUS topology: In this each computer is directly connected to primary network cable in a single line.
Advantages: Inexpensive, easy to install, simple to understand, easy to extend.

STAR topology: In this all computers are connected using a central hub.
Advantages: Can be inexpensive, easy to install and reconfigure and easy to trouble shoot physical problems.

RING topology: In this all computers are connected in loop.
Advantages: All computers have equal access to network media, installation can be simple, and signal does not degrade as much as in other topologies because each computer regenerates it.

61. What is mesh network?
A network in which there are multiple network links between computers to provide multiple paths for data to travel.

62. What is difference between baseband and broadband transmission?
In a baseband transmission, the entire bandwidth of the cable is consumed by a single signal. In broadband transmission, signals are sent on multiple frequencies, allowing multiple signals to be sent simultaneously.

63. What is packet filter?
Packet filter is a standard router equipped with some extra functionality. The extra

functionality allows every incoming or outgoing packet to be inspected. Packets meeting some criterion are forwarded normally. Those that fail the test are dropped.

64. What is traffic shaping?
One of the main causes of congestion is that traffic is often busy. If hosts could be made to transmit at a uniform rate, congestion would be less common. Another open loop method to help manage congestion is forcing the packet to be transmitted at a more predictable rate. This is called traffic shaping.

65. What is multicast routing?
Sending a message to a group is called multicasting, and its routing algorithm is called multicast routing.

66. What is Kerberos?
It is an authentication service developed at the Massachusetts Institute of Technology. Kerberos uses encryption to prevent intruders from discovering passwords and gaining unauthorized access to files.

67. What is passive topology?
When the computers on the network simply listen and receive the signal, they are referred to as passive because they don't amplify the signal in any way. Example for passive topology - linear bus.

68. What are the advantages of Distributed Processing?
a. Security/Encapsulation
b. Distributed database
c. Faster Problem solving
d. Security through redundancy
e. Collaborative Processing

69. Name the factors that affect the reliability of the network?
a. Frequency of failure
b. Recovery time of a network after a failure

70. When a switch is said to be congested?
It is possible that a switch receives packets faster than the shared link can accommodate and

stores in its memory, for an extended period of time, then the switch will eventually run out of buffer space, and some packets will have to be dropped and in this state is said to congested state.

Software Testing Interview Questions

1. What's the Software Testing?
A set of activities conducted with the intent of finding errors in software.

2.What is Acceptance Testing?
Testing conducted to enable a user/customer to determine whether to accept a software product. Normally performed to validate the software meets a set of agreed acceptance criteria.

3. What is Accessibility Testing?
Verifying a product is accessible to the people having disabilities (deaf, blind, mentally disabled etc.).

4. What is Ad Hoc Testing?
A testing phase where the tester tries to 'break' the system by randomly trying the system's functionality.

5. What is Application Programming Interface (API)?

A formalized set of software calls and routines that can be referenced by an application program in order to access supporting system or network services.

6. What is Backus-Naur Form?

A metalanguage used to formally describe the syntax of a language.

7. What is Beta Testing?

Testing of a release of a software product conducted by customers.

8. What is Application Binary Interface (ABI)?

A specification defining requirements for portability of applications in binary forms across different system platforms and environments.

9. What is Binary Portability Testing?

Testing an executable application for portability across system platforms and environments, usually for conformation to an ABI specification.

10. What is Black Box Testing?

Testing based on an analysis of the specification of a piece of software without reference to its internal workings. The goal is to test how well the component conforms to the published requirements for the component.

11. What is Bottom Up Testing?

An approach to integration testing where the lowest level components are tested first, then used to facilitate the testing of higher level components. The process is repeated until the component at the top of the hierarchy is tested.

12. What is Boundary Testing?
Test which focus on the boundary or limit conditions of the software being tested. (Some of these tests are stress tests).

13. What is the difference between verification and validation?
Verification is a review without actually executing the process while validation is checking the product with actual execution. For instance, code review and syntax check is verification while actually running the product and checking the results is validation.

14. What is Bug?
A fault in a program which causes the program to perform in an unintended or unanticipated manner.

15. What is Defect?
If software misses some feature or function from what is there in requirement it is called as defect.

16. What is Branch Testing?
Testing in which all branches in the program source code are tested at least once.

17. What is Breadth Testing?
A test suite that exercises the full functionality of a product but does not test features in detail.

18. What's the Alpha Testing ?
The Alpha Testing is conducted at the developer sites and in a controlled environment by the end user of the software

19. What's the Beta Testing ?
Testing the application after the installation at the client place.

20. What is Component Testing ?

Testing of individual software components (Unit Testing).

21. What is End-to-End testing ?

Testing a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

22. What is CAST?

Computer Aided Software Testing.

23. What is CMM?

The Capability Maturity Model for Software (CMM or SW-CMM) is a model for judging the maturity of the software processes of an organization and for identifying the key practices that are required to increase the maturity of these processes.

24. What is Cause Effect Graph?

A graphical representation of inputs and the associated outputs effects which can be used to design test cases.

25. What is Coding?
The generation of source code.

26. What is Compatibility Testing?
Testing whether software is compatible with other elements of a system with which it should operate, e.g. browsers, Operating Systems, or hardware.

27. What is Cyclomatic Complexity?
A measure of the logical complexity of an algorithm, used in white-box testing.

28. What is Debugging?
The process of finding and removing the causes of software failures.

29. What is Dependency Testing?
Examines an application's requirements for pre-existing software, initial states and configuration in order to maintain proper functionality.

30. What are the different Ways of doing Black Box testing?
There are five methodologies most frequently used:
A)Top down according to budget
B)WBS (Work Breakdown Structure)
C)Guess and gut feeling
D)Early project data
E)TPA (Test Point Analysis)

31 What's the Database testing?
In database testing, we can check the integrity of database field values.

32. How many types of testing?
There are two types of testing-
Functional- Black Box Testing
Structural- white Box Testing

33. What does the mclabe cyclomatic complexity of a program determine?
Cyclomatic complexity is likely the most widely used complexity metric in software engineering. It describes the complexity of a procedure by measuring the linearly independent paths through its source code.

34. What is the difference between interoperability and compatibility testing with some examples?
Interoperatability:-To check if the software can co exist with other supporting softwares in the system
Compatibility:-To check if the software runs on different types of operating systems according to customer requirements.

35. Which testing method is used to check the software in abnormal condition?

1) Stress testing
2) Security testing
3) Recovery testing
4) Beta testing

36. What's the Test Case?
A set of test inputs, execution, and expected result developed for a particular objective.

37. What's the Traceability Matrix?
A document that showing the relationship between Test Requirements and Test Cases.

38. How many types of approaches are used in Integration Testing?
There are two types of approaches used-

Bottom-Up
Top-Down

39. What is Emulator?

A device, computer program, or system that accepts the same inputs and produces the same outputs as a given system.

40. What is Functional Decomposition?

A technique used during planning, analysis and design; creates a functional hierarchy for the software.

41. What is Glass Box Testing?
A synonym for White Box Testing.

42. What is Gorilla Testing?
Testing one particular module, functionality heavily.

43. What is Gray Box Testing?

A combination of Black Box and White Box testing methodologies testing a piece of software against its specification but using some knowledge of its internal workings.

44. What is Integration Testing?
Testing of combined parts of an application to determine if they function together correctly. Usually performed after unit and functional testing. This type of testing is especially relevant to client/server and distributed systems.

45. What is Metric?

A standard of measurement. Software metrics are the statistics describing the structure or content of a program. A metric should be a real objective measurement of something such as number of bugs per lines of code.

46. What is Quality Assurance?
All those planned or systematic actions necessary to provide adequate confidence that a product or service is of the type and quality needed and expected by the customer.

47. What is Quality Control?
The operational techniques and the activities used to fulfill and verify requirements of quality.

48. What is Race Condition?
A cause of concurrency problems. Multiple accesses to a shared resource, at least one of which is a write, with no mechanism used by either to moderate simultaneous access.

49. What is Scalability Testing?
Performance testing focused on ensuring the application under test gracefully handles increases in work load.

50. What is Software Requirements Specification?
A deliverable that describes all data, functional and behavioral requirements, all constraints, and all validation requirements for software.

HR Interview Questions

As a fresher, you often experience a lot of anxiety before you go to attend an interview. The solution to this anxiety is good preparation. Here are some interview questions which you can expect to be asked during your HR interview with a way to answer them:

1. Did you face any difficulty in finding this place?

The interviewer usually asks this question to initiate the talk and put the candidate to ease. You can smile and answer this question by saying: “It wasn’t difficult to find this place.”

2. Tell me something about yourself.

This is the most commonly asked first question in any interview and the one which can set a ground for rest of your interview. Include details like: Your education background (not beyond 12th standard), some information about your family – father’s profession, mother’s profession, number of siblings (no need to get into their educational details) and some information about your extracurricular activities or hobbies.

3. Why does this role interest you? Or why have you applied for this job?

The purpose of this question is to see if you are really interested in this job or you applied to this job just because you are jobless. Relate the job requirements with your candidature and explain why you believe you are a suitable candidate for this job.

4. Why do you think should we take you for this job?

This is another way of asking the last question. A company will want to hire you if you can add some value to it. So, focus your answer on your skills and how they can be helpful for the position you are being interviewed for. Also, tell them about your ability to grasp new things quickly, adjust well into a team and flexible attitude. These are the qualities that employers usually seek in a fresh graduate.

Remember to keep an example ready for each personal quality you offer.

5. Are you an Introvert or Extrovert?

Being on either of the extremes at work can be bad for you and the company. Your behavior at work should be as per the requirement. To answer this question you can say something like: I behave as per the requirement of the situation. For e.g. If I am representing the company at an event which requires me to talk to a lot of people, I become an extrovert while if there is some serious issue under discussion, I think quietly and deeply on it.

6. What is your greatest strength?

This is also one of the most commonly asked questions in any interview. Your answer to this question demonstrates your preparation for the interview. Your same set of strengths can not be an advantage in every interview. This needs you to tailor make your answer as per the requirement of the position you are being interviewed for.

Before every interview, analyze the requirements of the role carefully and list out the qualities required to perform that role. Now, make a list of the qualities you have and match them with the requirements. You can offer these qualities as your strength for the role.

7. What do you want to achieve in the next 5 years?

This answer shows your ability to make plans for yourself, have an ambitious but realistic plan:
• Say you would like to acquire certain skills and move up the ladder
• If you are interested in Management, say you would like to take up a part-time course/full time course in x years time

8. Are you willing to change your project/profile when required?

This is very important for a company, they want employees who are willing to change roles, to meet business demands, so the typical answer here is "Yes". Say "I understand it will be in best interest of the business, if my manager wants me to develop my skills and take up another role."

9. What skills do you want to develop to succeed in this role?

Again talk about some key skills that are necessary for the role, this will show them that you are ready to learn and are aware of your shortcomings:
• Tell them that you, as a fresher, have the ability but formal training to enhance your skills for a professional environment

10. Are you planning to go for further studies?

This answer needs to be answered tactfully, they want to know whether you are going to stay for long or you are just there to use their training resources and then say goodbye in 6 month's time.
• If you are pursuing further studies, say so. Tell them why you want to go for that course
• If you are taking up a distance education course or a part time course, they should know, as you will need to take leaves when you appear for exams

11. What newspaper/magazine/book do you read?

• Browse through the newspaper that morning they might ask you what was the front page headline, they might ask you your favorite columnists name
• Same goes for the magazine, make sure you have read the last edition
• When giving names of books, remember the author(s) name(s) and be ready to give out some summary of the book

12. Introduce yourself

This question is usually the first question they HR might ask you. This question could form a basis for further discussion.
• Discuss your family background in short
• Discuss your qualification
• Discuss your characteristics that will help you professionally
• Discuss your interest, hobbies

13. Tell me something about yourself that is no written in your resume.

If you missed out on some points while writing your resume this is your chance to discuss them:
• Whenever talking about yourself, be confident, but not over confident.
• Do not praise yourself all the time.
• You could discuss some personal traits or achievements

14. Do you know anyone who works with this company?

It is fine if you do not know anyone, but if you know it will be good to give out the name:
• This shows them about your level of interest, and it is human nature if you know someone who works there you will ask them for feedback about the company or the role
• They could ask that person for your reference

15. Do you have any other offer at hand?

This is usually a closing question and they might want to know how actively you are looking for a job. Usually give an honest answer and say why you are looking for another job with that offer at hand.

16. Do you have any questions for us?

Always have a question ready to answer this one:
• You can ask whether the company allows for lateral and vertical role changes
• You can also ask whether the company encourages learning and development of employees
• Ask whether the company has plans for expansion
• You can also discuss your role in detail.

⋆-⋆-⋆_Best_Of_Luck_⋆-⋆-⋆