# First Round Common Questions

1. Draw the high level architecture of your project? **(2 mins)**

**Follow up questions (5 minutes)**

* 1. Is it a 2 tier or 3 tier architecture?
  2. If the application in the project doesn’t has service layer and there is a possibility for adding that, then ask him/her about how would you enhance the existing application by introducing a service layer?
  3. How did you test your application? Your code?
  4. What was your role in this project? What was the team size?
  5. What all modules you have single handedly written/coded?
     1. How many classes? How many functions? How long was the biggest function? Why?
  6. What was the most difficult code you implemented as part of this?
  7. What was the most difficult issue you debugged?
  8. What is the IDE that you have used?
  9. Did you use any source version control?
  10. What data structures did you use on the project?
  11. What language features did you learn as part of the project which were not in your curriculum?

1. What is object oriented design?
   1. What are SOLID design principles?

# Level 1

OOPS –

1. What is OOPS? What are the advantages?
2. What is the difference between OOPS and procedural programming?
3. What is difference between Abstraction and Encapsulation?
4. What is inheritance? Difference between multiple and multi-level ?
5. What is Static binding and dynamic binding?
6. What is the difference between method overloading and overriding?
7. What is association and dependency? Difference ?
8. Where do we use polymorphism ? What are the types of polymorphism?
9. What are different types of constructors ?
10. What all access specifiers you know ?
11. What is Enum ?
12. What is hashtable ?
13. What is garbage collector?
14. What is an Interface?
15. What is Exceptional handling? Example ?
16. What is difference between Static and Const ?
17. What is This pointer ?
18. What is a Partial Class ?
19. What is a event?
20. Int a = 29;   
    a--;  
     a -= ++a;  
      
    Print(a);  
      
    Output: -1

C programming-

Level 1-

1.What is a variable? What are local and global variables?

Ans. Variable is a tool to reserve space in computer’s memory. Reserved space is given a name which we call a variable name and constant are stored in this reserved space.

2. What is a ternary operator?

Ans. Exp1 ? Exp 2 : Exp 3

3. What is the difference between BREAK and CONTINUE?

Ans: Continue cannot be used with Switch cases.

4. What is the difference between switch and if-else ?

Ans. Logical operators or float cannot be used in cases for switch.

5. What is a function? What is a recursive function ?

Ans. A function is a self- contained block of code that performs a coherent task of some kind.

6. What is the difference call by value and call by reference ? with example.

7. What does ‘#include <stdio.h>’ means?

8. What is the difference between #include <..> and #include ”..”?

Ans. Double quote mark means that the directory you are currently in, will be searched first, for the header file, before any other directory.  
 Angled brackets – dir other than the one you are currently in, will be searched for the header file. The system dependent dirs. Are searched for the header files.

9. What is a Class, Structure and a Union. Difference between them?

10. What is LValue and RValue ?

Ans. Expressions that refer to memory locations are called LValue. The R-Value is the data value of the variable.

Int a ;

a= 5;

5 is a rvalue, a is a lvalue.

11. What is the difference between Malloc and Calloc function?

12. What is the difference between strcpy() and memcpy() function ?

Ans. Strcpy- copies char in the source string to the destination till it reaches string’s NULL char.  
 memcpy – copies char in the source to the destination string till the given length.

13. What is the difference between unsigned and signed int?

14. 8. How many types of storage classes are there ?

Ans. Automatic, register, static and External

15. What is a Preprocessor ? How many types ?

Ans. Preprocessor is a program that processes our source program before passing it on for compilation.   
Types- Macro Expansion, File Inclusion and Conditional Compilation.

C++ –

1. What is friend function? Explain with a code example.
2. What is copy constructor? Explain with code example.
3. What is operator overloading? Explain with code example.
4. What is inline function? How it is different from C Macro?
5. Mention the use of scope resolution operator with code example.
6. Difference between Pointer and reference ?
7. What is memory leak? How will you plug these leaks?
8. What is shallow and deep copy ?

JAVA –

1. What is the difference between Java Servlet and Java applet?
2. What is Java bean ?
3. What is Bytecode? Is Java platform dependent or independent? What makes Java platform independent? Draw the architecture of Java code compilation.
4. What is the use of InstanceOf operator? Small code snippet for example.
5. What are jar, war and ear files ?
6. What is JIT compiler ?
7. What is the difference between string and stingbuffer class ?
8. What is a thread life cycle? Explain with Example. What is difference between sleeping and yielding in Threads ?
9. What is the difference between notify() and notifyall() in threading ?
10. What are the differences between final, finally and finalize ?

Data Structure and Algo-

1. What is data structure?
2. What is an algorithm?
3. Explain the algorithm used to merge two sorted arrays.
4. Explain the procedure to insert into a sorted array.
5. What is STACK and Queue? Explain their basic operations.
6. What is linked list? How many types of linked list are there, explain them?
7. Explain linear search in a sorted array.

Database-

1. What is the difference between DBMS and RDBMS ?
2. What is referential integrity?
3. What are E-R diagrams ? Give an example.
4. What is the difference between primary key and unique key?
5. What is normalization? Explain the forms of normalization.
6. What is DDL and DML? Explain commands used in both.
7. What is transaction and ACID properties?
8. What are constraints? What are different types of constraints?
9. What are triggers?
10. What is a Stored Procedure? Difference between trigger and stored procedure ?
11. What is cascading update?
12. Explain the difference between an exclusive lock and shared lock ?
13. What do you mean by flat file database?
14. What is indexing? What are the different kinds of indexing?
15. What is lossless join property?
16. What is extension and intension ?
17. What is Cursor?
18. What is sub-query?
19. What is Data warehouse?
20. What is the difference between HAVING CLAUSE and WHERE CLAUSE ?

Software Engineering-

1. Explain any software development lifecycle.
2. Explain PDCA cycle. (Plan Do Check Act) and where testing fits in?
3. What is difference between white box, black box and grey box testing?
4. What are the categories of defects?   
     
   Ans- Wrong, missing and extra.
5. What is difference between a defect and a failure?  
     
   Ans. When a defect reaches the end customer it is called failure.
6. What all phases of testing are performed during a software development life cycle ?  
     
   Ans. unit, integration, system and acceptance.
7. What is difference between verification and validation?
8. What is difference between authorization and authentication?
9. What is difference between authorization and authentication?
10. What is difference between alpha and beta testing?
11. What is regression testing?
12. What is a network?
13. What is a protocol?
14. Define Routing.
15. What is Round Trip Time ?
16. Define unicasting, multicasting and broadcasting?
17. What is Error Detection?
18. What is Piggy backing?
19. What are the different data transmission types?  
      
    Ans. Simplex, duplex, half duplex.
20. What is difference between gateway and router?
21. What is silly window syndrome?
22. What is cohesion and coupling?
23. What is main difference between a computer program and computer software?
24. What is modularization?
25. What is UML? Explain with example.
26. What is Quality Assurance and Quality Control ?
27. What is difference between a Exe and a dll ?
28. What is difference between assembly and namespace?
29. What is strong-typing and weak typing?
30. What is Race condition?
31. Different datatypes they know.What will be appropriate for a given type of input.e.g when considering distance value, population etc.
32. What types can be safely interchanged.
33. Given a positive integer N. Write a function to calculate sum of its digits. Write a class and call it from main method.
34. Write a function to alphabetically sort given strings. Call it from main method.
35. Write a SQL statement to.

Create a new table

Add 3 new rows

Remove last 2 rows.

1. Explain with code sample: difference between composition and aggregation?
2. Explain with code sample: What is the difference between interface and abstract class?
   1. How many instances can be created for an abstract class?
3. Explain with code sample: Overloading.
   1. Can a destructor be overloaded?
4. Explain with code sample: What are value types and reference types in C#?
   1. Is Int32 a value type or reference type in c#?
5. Can we have multiple inheritance in c#?
   1. Do you know any issues caused by having multiple inheritance?
   2. Do you know of any language that supports MI?
   3. How does the language address the issue?

The "diamond problem" (sometimes referred to as the "deadly diamond of death") is an ambiguity that arises when two classes B and C inherit from A, and class D inherits from both B and C. If there is a method in A that B and C have overridden, and D does not override it, then which version of the method does D inherit: that of B, or that of C?

C++ by default follows each inheritance path separately, so a D object would actually contain two separate A objects, and uses of A's members have to be properly qualified. If the inheritance from A to B and the inheritance from A to C are both marked "virtual" (for example, "class B : virtual public A"), C++ takes special care to only create one A object, and uses of A's members work correctly. If virtual inheritance and nonvirtual inheritance are mixed, there is a single virtual A, and a nonvirtual A for each nonvirtual inheritance path to A. C++ requires stating explicitly which parent class the feature to be used is invoked from i.e. "Worker::Human.Age". C++ does not support explicit repeated inheritance since there would be no way to qualify which superclass to use (i.e. having a class appear more than once in a single derivation list [class Dog : public Animal, Animal]). C++ also allows a single instance of the multiple class to be created via the virtual inheritance mechanism (i.e. "Worker::Human" and "Musician::Human" will reference the same object).

1. Explain with code sample: The difference between new and override in C#?
2. Explain with code sample: A Car is a Vehicle. What OOP concept this statement indicates?
3. Explain with code sample: Write a function to reverse a string without using built in functions.

Answer: There are several ways to do this. But a couple of good ones below.

Using recursion: Each call would reduce the input string by 1 character. And increase the output string by one character.

Using a char array: Create a single array. Swap elements. (traverse only once)

1. In an array 1-100 numbers are stored; one number is missing how do you find it?
2. Swap two numbers without using temp variable.
3. Get second highest number in integer array.
   1. Are there any algorithms involved in this?
4. Write a function to find the maximum consecutive 1's in an array of 0's and 1's.

   Example:

    a) 00110001001110 - Output :3 [Max num of consecutive 1's is 3]

    b) 1000010001 - Output :1 [Max num of consecutive 1's is 1]

Put the function in a class and call it from main.

1. What is result of following program

double GetValue (int a, int b) {

   return (a/b);

}

int main () {

   int x = 50;

   int y = 10;

   int z= 0;

   double t = 0;

     t = GetValue(x, y);

      cout << t << endl;

     t = GetValue(x, z);

      cout << t << endl;

   }

1. Fill in the blanks.



1. sin θ = \_\_\_\_\_\_\_\_\_\_

2. cos θ = \_\_\_\_\_\_\_\_\_

3. tan θ =  \_\_\_\_\_\_\_\_\_

4. sinθ / cosθ = \_\_\_\_\_\_\_

5. Given h and w,

l = \_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. You have a circle of radius r. center of circle is C. how could you decide whether point P is within the circle or outside it.
2. What is the ouput?

#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

1. Differentiate between a deep copy and a shallow copy?

***Answer:***

Deep copy involves using the contents of one object to create another instance of the same class. In a deep copy, the two objects may contain ht same information but the target object will have its own buffers and resources. the destruction of either object will not affect the remaining object. The overloaded assignment operator would create a deep copy of objects.

Shallow copy involves copying the contents of one object into another instance of the same class thus creating a mirror image. Owing to straight copying of references and pointers, the two objects will share the same externally contained contents of the other object to be unpredictable.

***Explanation:***

Using a copy constructor we simply copy the data values member by member. This method of copying is called shallow copy. If the object is a simple class, comprised of built in types and no pointers this would be acceptable. This function would use the values and the objects and its behavior would not be altered with a shallow copy, only the addresses of pointers that are members are copied and not the value the address is pointing to. The data values of the object would then be inadvertently altered by the function. When the function goes out of scope, the copy of the object with all its data is popped off the stack.

If the object has any pointers a deep copy needs to be executed. With the deep copy of an object, memory is allocated for the object in free store and the elements pointed to are copied. A deep copy is used for objects that are returned from a function.

1. class a

{

public:

 a(){printf("\nBase Constructor\n");}

~a(){printf("\nBase Destructor\n");}

 };

class b : public a

 {

public:

b(){printf("\nDerived Constructor\n");}

~b(){printf("\nDerived Destructor\n");}

 };

int main()

{

a\* obj=new b;

delete obj;

return 0;

}

**What is Output?**

Output:

Base Constructor

Derived Constructor

Base Destructor

**Follow up question A : Why Destructor of derived class is not getting called?** (Destructor of derived class should get called just before Destructor of base class.)

**OR**

**Follow up question B : What change I have to make to get called Destructor of derived class?**

1. Anything wrong in this c++ code

T\* x = 0;

delete x;

1. Which constructor of Class A is called for below function with given signature

Foo(A& a);

1. Will this program works fine?

#include <iostream>

using namespace std;

class Test

{

private:

   ~Test() {}

};

int main()

{

   Test \*t;

}

Answer : yes, There is no object being constructed, the program just creates a pointer of type “Test \*”, so nothing is destructed.

1. In below code what is value of I, j.

Int I =12;

Int j = i++;

Answer- I = 13. J = 12. If declared int j = ++I; then j will become 13

# Level 2

1. Write a function to create a number pyramid. The function should take an input to decide the height of the pyramid. Call the function from main method.
2. 10 commonly used data structures ? Can ask below questions as follow up.
3. What is binary search? What is the time complexity of this algorithm?

Binary search works on sorted arrays. Binary search begins by comparing the middle element of the array with the target value. If the target value matches the middle element, its position in the array is returned. If the target value is less than or greater than the middle element, the search continues in the lower or upper half of the array, respectively, eliminating the other half from consideration.

Time complexity: O(log n)

1. Implement the stack class?
2. You are given an array of integers (with all valid input) You have to write a function which will produce another array, where the value in each index of the array will be the product of all values in the given array accept that index.   
   Example   
   Array 1: 1 2 3 4 5   
   Array 2: 120 60 40 30 24.
3. Write a function: We have an array of objects A and an array of indexes B. Reorder objects in array A with given indexes in array B. Do not change array A's length.   
     
   example:

var A = [C, D, E, F, G];

var B = [3, 0, 4, 1, 2];

// A is now [D, F, G, C, E];

Put the function in a class and call it from main.

1. Write a function: Print all permutation of String both iterative (e.g. for loops) and Recursive way? Put the function in a class and call it from main.
2. Assume Family tree (e.g. Grandfather->3 sons & 2 daughters-> and so on) up to multiple level. Each member is represented by object of class FamilyNode. Class FamilyNode has two methods as follows.

Method # 1.       int NumberOfChildren FamilyNode::GetImmediateChildren (FamilyNode\* ImmediateChildren);

Method # 2.       void FamilyNode::PrintName ();

**Is this information enough to print names of all family members?**

Yes.

**Follow up question A : How Many line of code is required according to you?**

Less than 10.

**Follow up question B : Please write Pseudocode?**

Answer is Recursive Function.

**Follow up question C : In Recursive programming logic, how the function calls are managed?**

Function calls are managed on Last In First out basis. Recursive function is called multiple times. Each time next call is made to recursive function, current executing call is put on stack with it’s whole state. Then next call is being processed further. Then goes on till last call reaches exit condition. Once last call is complete. Then previous call on stack is being executed with it’s saved state. This continues till not calls left in the stack.

1. In a room of 12 \* 10 \* 10 (height). How to calculate a distance from one of the corner in ceiling to the center of the room on floor.
2. How many different ways you can take a stair? Constrain you can take maximum 2 steps at a time. E.g If we have a stair with 4 steps then few combination (1,1,1,1)(1,2,1,)(2,1,1)(2,2)(1,1,2)
3. Given a String of length **S**, reverse the whole string without reversing the individual words in it. Words are separated by **dots**. E.g i/p this.is.the.test o/p test.the.is.this
4. Design a parking lot.
   1. Expect to come up with class diagram categorizing the vehicle, capacity, parking level, special consideration (handicapped etc.)

Operating Systems-

1. What is an Operating System ?
2. What is a kernel ?
3. What is a process? What is the difference between a Process and a Thread ?
4. What is the difference between a compiler and an interpreter ?
5. What is a library ?
6. What is the difference between an assembly and a namespace ?
7. What is a Distributed system ?
8. What do you mean by 64-bit or 32-bit CPU ?
9. What is pagination ?
10. What is the difference between synchronous and asynchronous function ?