C Interview Questions:

1: Please explain what is a header file in C? What will happen if we include a header file twice in a C program?

Answer: Header files store the definitions and set of rules governing different built-in functions of the C programming language. For instance, the printf() and scanf() functions are defined in the stdio.h header file.

Every header file contains a set of predefined functions, meant to make programming in C simpler. You need to include the specific header file in your C program to be able to use the functions defined in it. For example, you can't use printf() and scanf() functions without including the stdio.h header file.

When a header file is included twice in a C program, the second one gets ignored. In actual, the #, called the include guard, preceding a header file ensures that it is included only once during the compilation process.

2: Please compare static memory allocation with dynamic memory allocation?

Answer: Following are the important differences between static and dynamic modes of memory allocation:

Memory increase:

In dynamic memory allocation, memory can be increased while executing the program. This is not the case; however, with the static memory allocation where the option of increasing memory during program execution is not available.

Memory requirement

Static memory allocation needs more memory space compared to dynamic memory allocation.

Used in

Arrays use static memory allocation while linked lists use dynamic memory allocation.

When does it happen?

Static memory allocation takes place at compile-time, while dynamic memory allocation happens at runtime.

3: Can you explain memory leak in C? Why should it be addressed?

Answer: Memory leak happens when a memory created in a heap remains undeleted. This can lead to additional memory usage and, thus, affect the performance of a program. This is exactly why the issue of memory leak must be addressed.

4: Please explain what do you understand by while(0) and while(1)?

Answer: while(0) means that the looping conditions will always be false, i.e., the code inside the while loop will not be executed. On the opposite, while(1) is an infinite loop. It runs continuously until coming across a break statement mentioned explicitly.

Note: Any non-zero integer inside the braces of the while loop will give an infinite loop. For example, while(-22) and while(24) will both yield an infinite loop.

5: Explain the purpose of the 'delete' operator?

Answer: Delete removes all the objects created by the new expression, i.e. frees memory in the heap space. The array objects are deleted using the [] operator:

delete[] array;

NULL or void Pointer can be deleted as:

delete ptr;

The same is applicable for user-defined data types as well. For example,

int *var = new int;

delete var;

C++ Interview Questions:

1:What is the precedence when there are a Global variable and a Local variable in the program with the same name?

Answer: Whenever there is a local variable with the same name as that of a global variable, the compiler gives precedence to the local variable.

Example:

```
#include <iostream.h>

 int globalVar = 2;

int main()

{
```

```
 int globalVar = 5;

 cout<<globalVar<<endl;

}
```

The output of the above code is 5. This is because, although both the variables have the same name, the compiler has given preference to the local scope.

2: What are the Extraction and Insertion operators in C++? Explain with examples.

Answer: In the iostream.h library of C++, cin, and cout are the two data streams that are used for input and output respectively. Cout is normally directed to the screen and cin is assigned to the keyboard.

“cin” (extraction operator): By using overloaded operator >> with cin stream, C++ handles the standard input.

```
int age;

cin>>age;
```

As shown in the above example, an integer variable ‘age’ is declared and then it waits for cin (keyboard) to enter the data. “cin” processes the input only when the RETURN key is pressed.

“cout” (insertion operator): This is used in conjunction with the overloaded << operator. It directs the data that followed it into the cout stream.

Example:

```
cout<<”Hello, World!”;

 cout<<123;
```

3: What is the difference between an Object and a Class?

Answer: Class is a blueprint of a project or problem to be solved and consists of variables and methods. These are called the members of the class. We cannot access methods or variables of the class on its own unless they are declared static.

In order to access the class members and put them to use, we should create an instance of a class which is called an Object. The class has an unlimited lifetime whereas an object has a limited lifespan only.

4: What is a COPY CONSTRUCTOR and when is it called?

Answer: A copy constructor is a constructor that accepts an object of the same class as its parameter and copies its data members to the object on the left part of the assignment. It is useful when we need to construct a new object of the same class.

Example:

```
class A{
        int x; int y;
        public int color;
        public A() : x(0) , y(0) {} //default (no argument) constructor
        public A( const A& ) ;
};
A::A( const A & p )
{
         this->x = p.x;
         this->y = p.y;
         this->color = p.color;
}
main()
{
       A Myobj;
       Myobj.color = 345;
       A Anotherobj = A( Myobj ); // now Anotherobj has color = 345
}
```

5: What is the difference between a Copy Constructor and an Overloaded Assignment Operator?

Answer: A copy constructor and an overloaded assignment operator basically serve the same purpose i.e. assigning the content of one object to another. But still, there is a difference between the two.

Example:

```
complex c1,c2;
c1=c2; //this is assignment
complex c3=c2; //copy constructor
```

In the above example, the second statement c1 = c2 is an overloaded assignment statement.

Here, both c1 and c2 are already existing objects and the contents of c2 are assigned to the object c1. Hence, for overloaded assignment statement both the objects need to be created already.

Next statement, complex c3 = c2 is an example of the copy constructor. Here, the contents of c2 are assigned to a new object c3, which means the copy constructor creates a new object every time when it executes.