**Interview Details –** First round was a hr behavioral round. They asked me questions like,  
1) How do you resolve a conflict in your team?(highly asked)  
2) have you even been a team leader ?and similar kind of questions.

Second round of technical, I had chosen digital signal processing.  
Math questions.  
1) What is a singular matrix?

* A square matrix that is not invertible is called **singular (or its rank is less than its dimension)**
* The **column rank** of a matrix *A* is the maximum number of linearly independent column vectors of *A*. The **row rank** of a *A* is the maximum number of linearly independent row vectors of *A*. Equivalently, the column rank of *A* is the [dimension](http://en.wikipedia.org/wiki/Dimension_(linear_algebra)) of the [column space](http://en.wikipedia.org/wiki/Column_space) of *A*, while the row rank of *A* is the dimension of the [row space](http://en.wikipedia.org/wiki/Row_space)of *A*.

2) what is a cross correlation?

* **cross-correlation** is a measure of similarity of two random variables.
* In [probability theory](http://en.wikipedia.org/wiki/Probability_theory) and [statistics](http://en.wikipedia.org/wiki/Statistics), *correlation* is always used to include a standardising factor in such a way that correlations have values between −1 and +1, and the term **cross-correlation** is used for referring to the [correlation](http://en.wikipedia.org/wiki/Covariance_and_correlation) corr(*X*, *Y*) between two [random variables](http://en.wikipedia.org/wiki/Random_variables) *X* and *Y*, while the "correlation" of a random vector *X* is considered to be the [correlation matrix](http://en.wikipedia.org/wiki/Correlation_matrix) (matrix of correlations) between the scalar elements of*X*.
* In [time series analysis](http://en.wikipedia.org/wiki/Time_series_analysis), as applied in [statistics](http://en.wikipedia.org/wiki/Statistics), the cross correlation between two time series describes the normalized cross covariance function

3) what is random process?

* **random process** is a collection of [random variables](http://en.wikipedia.org/wiki/Random_variable); this is often used to represent the evolution of some random value over time.
* stochastic process amounts to a [sequence](http://en.wikipedia.org/wiki/Sequence_(mathematics)) of random variables known as a [time series](http://en.wikipedia.org/wiki/Time_series)

4) what is eigen value of this matrix?  
5) what is eigen vector of this matrix?

* An **eigenvector** of a [square matrix](http://en.wikipedia.org/wiki/Matrix_(mathematics)) A is a non-zero [vector](http://en.wikipedia.org/wiki/Vector_(mathematics)) v that, when [multiplied](http://en.wikipedia.org/wiki/Matrix_multiplication) by A, yields the original vector multiplied by a single number \lambda; that is, A v = \lambda v. The number \lambda is called the **eigenvalue** of A corresponding to v

6) inverse of a matrix for a 2\*2 matrix?   
7) what is a symmetric matrix?

* In [linear algebra](http://en.wikipedia.org/wiki/Linear_algebra), a **symmetric matrix** is a [square matrix](http://en.wikipedia.org/wiki/Square_matrix) that is equal to its [transpose](http://en.wikipedia.org/wiki/Transpose), i.e. a\_ij=a\_ji

8) what is dy/dx?  
9) what is partial differentiation?

* In [mathematics](http://en.wikipedia.org/wiki/Mathematics), a **partial derivative** of a [function](http://en.wikipedia.org/wiki/Function_(mathematics)) of several variables is its [derivative](http://en.wikipedia.org/wiki/Derivative) with respect to one of those variables, [with the others held constant](http://en.wikipedia.org/wiki/Ceteris_paribus" \o "Ceteris paribus) (as opposed to the [total derivative](http://en.wikipedia.org/wiki/Total_derivative), in which all variables are allowed to vary)
* A partial differentiation of a multivariable [function](http://en.wikipedia.org/wiki/Function_(mathematics)) is its [derivative](http://en.wikipedia.org/wiki/Derivative) with respect to one of those variables [with the others held constant](http://en.wikipedia.org/wiki/Ceteris_paribus)

10) what is diff between differential eqn and partial diff eqn? (ODE v.s PDE)

* In [mathematics](http://en.wikipedia.org/wiki/Mathematics), an **ordinary differential**[**equation**](http://en.wikipedia.org/wiki/Ordinary_differential_equations) (abbreviated **ODE**) is an equation containing a function of one [independent variable](http://en.wikipedia.org/wiki/Independent_variable) and its derivatives
* In [mathematics](http://en.wikipedia.org/wiki/Mathematics), a **partial differential equation** (**PDE**) is a [differential equation](http://en.wikipedia.org/wiki/Differential_equation) that contains unknown [multivariable functions](http://en.wikipedia.org/wiki/Multivariable_calculus) and their [partial derivatives](http://en.wikipedia.org/wiki/Partial_derivatives)(This is in contrast to [ordinary differential equations](http://en.wikipedia.org/wiki/Ordinary_differential_equations), which deal with functions of a single variable and their derivatives)

11) Given acceleration find velocity?

* **v = v0 + at, given a, a=dv/dt, integral v**
* **acceleration** is the [rate](http://en.wikipedia.org/wiki/Rate_(mathematics)) at which the [velocity](http://en.wikipedia.org/wiki/Velocity) of a body changes with time

12) what is auto correlation?

* **Autocorrelation** is the [cross-correlation](http://en.wikipedia.org/wiki/Cross-correlation) of a [signal](http://en.wikipedia.org/wiki/Signal_(information_theory)) with itself
* Autocorr()

13) what is a null matrix?

* a **zero matrix** or **null matrix** is a [matrix](http://en.wikipedia.org/wiki/Matrix_(mathematics)) with all its [entries](http://en.wikipedia.org/wiki/Zero_matrix) being [zero](http://en.wikipedia.org/wiki/0_(number)" \o "0 (number))Z

14) how do you find a rank of a matrix?  
15) what is integration?

* Given a [function](http://en.wikipedia.org/wiki/Function_(mathematics)) *f* of a [real](http://en.wikipedia.org/wiki/Real_number) [variable](http://en.wikipedia.org/wiki/Variable_(mathematics)) *x* and an [interval](http://en.wikipedia.org/wiki/Interval_(mathematics)) [*a*, *b*] of the [real line](http://en.wikipedia.org/wiki/Real_line), the **definite integral**

\int_a^b \! f(x)\,dx \,

is defined informally to be the [area](http://en.wikipedia.org/wiki/Area_(geometry)) of the region in the *xy*-plane bounded by the [graph](http://en.wikipedia.org/wiki/Graph_of_a_function) of *f*, the *x*-axis, and the vertical lines*x* = *a* and *x* = *b*, such that area above the *x*-axis adds to the total, and that below the *x*-axis subtracts from the total.

The term *integral* may also refer to the notion of the [antiderivative](http://en.wikipedia.org/wiki/Antiderivative" \o "Antiderivative), a function *F* whose [derivative](http://en.wikipedia.org/wiki/Derivative) is the given function *f*. In this case, it is called an *indefinite integral* and is written:

F = \int f(x)\,dx.

16) what is differentiation?

* Differentiation is a process to calculate rate or change
* the **derivative** is a measure of how a [function](http://en.wikipedia.org/wiki/Function_(mathematics)) changes as its [input](http://en.wikipedia.org/wiki/Derivative) changes. Loosely speaking, a derivative can be thought of as how much one quantity is changing in response to changes in some other quantity; for example, the derivative of the position of a moving object with respect to time is the object's instantaneous [velocity](http://en.wikipedia.org/wiki/Velocity)

17) what is derivative?  
18) relation between differentiation and integration?

* They are a pair of inverse opertions

19) gave me sums on partial differential equation?

Java

3) Garbage collection in java.

### 1) objects are created on[heap in Java](http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html)  irrespective of there scope e.g. local or member variable. while its worth noting that class variables or static members are created in method area of [Java memory space](http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html) and both heap and method area is shared between different thread. 2) Garbage collection is a mechanism provided by Java Virtual Machine to reclaim [heap space](http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html) from objects which are eligible for Garbage collection. 3) Garbage collection relieves java programmer from memory management which is essential part of C++ programming and gives more time to focus on business logic. 4) Garbage Collection in Java is carried by a daemon thread called *Garbage Collector*. 5) Before removing an object from memory Garbage collection thread invokes finalize () method of that object and gives an opportunity to perform any sort of cleanup required. 6) You as Java programmer can not force Garbage collection in Java; it will only trigger if JVM thinks it needs a garbage collection based on [Java heap size](http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html). 7) There are methods like System.gc () and Runtime.gc () which is used to send request of Garbage collection to JVM but it’s *not guaranteed that garbage collection will happen*. 8) If there is no memory space for creating new object in Heap Java Virtual Machine throws OutOfMemoryError or [java.lang.OutOfMemoryError heap space](http://javarevisited.blogspot.com/2011/05/java-heap-space-memory-size-jvm.html) 9) J2SE 5(Java 2 Standard Edition) adds a new feature called Ergonomics goal of ergonomics is to provide *good performance from the JVM with minimum of command line tuning*.

### When an Object becomes Eligible for Garbage Collection

An Object becomes **eligible for Garbage collection or GC** if **its not reachable from any live threads or any static refrences** in other words you can say that an object becomes eligible for garbage collection if its *all references are null.* **Cyclic dependencies** are not counted as reference so if Object A has reference of object B and object B has reference of Object A and they don't have any other live reference then both Objects A and B will be ***eligible for Garbage collectio***n.   
Generally an object becomes *eligible for garbage collection in Java* on following cases:  
1) All references of that object explicitly set to null e.g. object = null  
2) Object is created inside a block and reference goes out scope once control exit that block.  
3) Parent object set to null, if an object holds reference of another object and when you set container object's reference null, child or contained object automatically becomes eligible for garbage collection.  
4) If an object has only live references via **WeakHashMap** it will be eligible for garbage collection. To learn more about HashMap see here [How HashMap works in Java](http://javarevisited.blogspot.com/2011/02/how-hashmap-works-in-java.html)

4) abstarct classes, 5)generics in java, constructors, #ifdef, #define,

6) Mutex, Semaphore, final variables,

dangling pointers

* **Dangling pointers** and **wild pointers** in [computer programming](http://en.wikipedia.org/wiki/Computer_programming) are [pointers](http://en.wikipedia.org/wiki/Data_pointer) that do not point to a valid object of the appropriate type. These are special cases of [memory safety](http://en.wikipedia.org/wiki/Memory_safety) violations.
* Dangling pointers arise when an object is deleted or deallocated, without modifying the value of the pointer, so that the pointer still points to the memory location of the deallocated memory. As the system may reallocate the previously freed memory to another process, if the original program then [dereferences](http://en.wikipedia.org/wiki/Dereference_operator) the (now) dangling pointer, [*unpredictable behavior*](http://en.wikipedia.org/wiki/Undefined_behavior)*may result*, as the memory may now contain completely different data. This is especially the case if the program writes data to memory pointed by a dangling pointer, a silent corruption of unrelated data may result, leading to subtle [bugs](http://en.wikipedia.org/wiki/Software_bug) that can be extremely difficult to find, or cause [segmentation faults](http://en.wikipedia.org/wiki/Segmentation_fault) (UNIX, Linux) or [general protection faults](http://en.wikipedia.org/wiki/General_protection_fault) (Windows). If the overwritten data is bookkeeping data used by the system's memory allocator, the corruption can cause system instabilities
* memory leak
* Let us start with outlining what is the difference in the memory management in Java and, for example, C languages. When a C-programmer wants to use a variable, he has to manually allocate a region in the memory where the value will reside. After the application finishes using that value, the region of the memory must be manually freed, i.e. the code freeing the memory has to be written by the developer. In Java, when a developer wants to create and use a new object using, e.g. new Integer(5), he doesn’t have to allocate memory – this is being taken care of by the Java Virtual Machine (JVM). During the life of the application JVM periodically checks which objects in memory are still being used and which are not. **Unused objects can be discarded and memory reclaimed and reused again**. This process is called **garbage collection** and the corresponding piece of JVM is called Garbage Collector, or GC.
* Java’s automatic memory management relies on GC which periodically looks for unused objects and removes them. And here hides the dragon. Simplifying a bit, **we can say that a memory leak in Java is a situation where some objects are not used by application any more, but GC fails to recognize** them as unused. As a result, these objects remain in memory indefinitely reducing the amount of memory available to the application

1) What is diff between jre and jdk?

* Java running environment
* Jave development kit

2) what is use of string and string buffer?

* Java provides the StringBuffer and String classes, and the String class is used to manipulate character strings that cannot be changed. Simply stated, objects of type String are read only and immutable. The StringBuffer class is used to represent characters that can be modified
* The significant performance difference between these two classes is that StringBuffer is faster than String when performing simple concatenations
* the most important difference between String and StringBuffer/StringBuilder in java is that **String object is immutable** whereas **StringBuffer/StringBuilder objects are mutable**

3) int and integer?

* **An int is a number; an Integer is a pointer that can reference an object that contains a number**
* In Java, the 'int' type is a primitive , whereas the 'Integer' type is an object.
* The differences between objects and primitives are somewhat beyond the scope of this question, but to summarize:
* **Objects** provide facilities for polymorphism, are passed by reference (or more accurately have references passed by value), and are allocated from the [heap](http://en.wikipedia.org/wiki/Dynamic_memory_allocation). Conversely, **primitives** are immutable types that are passed by value and are often allocated from the [stack](http://en.wikipedia.org/wiki/Stack-based_memory_allocation).

4) java [application](http://www.glassdoor.com/Interview/MathWorks-Interview-Questions-E17117.htm) vs java applet?

* Java applets and applications are similar, however distinctions do exist. Applets are generally shorter and are designed to run from Web pages. Java applets are frequently executed by clicking on an icon with a Web browser
* The simple distinction between Java applets and applications is that an applet is executed from within a browser and an Application is executed by the Java program
* **Applets as previously described, are the small programs while applications are larger programs. Applets don't have the main method while in an application execution starts with the main method. Applets can run in our browser's window or in an appletviewer**

5) break and continue.  
6) Pointers?  
7) why java is machine independent?

* Because all Java bytecode runs on the Java JVM, which in turn runs on top of the OS. Thus, Java separates the execution environment from the OS, making the Java code machine independent. The same Java code can be run on pretty much any machine, provided that a JVM is available

8) call by value and call by reference?

* **Objects** provide facilities for polymorphism, are passed by reference (or more accurately have references passed by value), and are allocated from the [heap](http://en.wikipedia.org/wiki/Dynamic_memory_allocation). Conversely, **primitives** are immutable types that are passed by value and are often allocated from the [stack](http://en.wikipedia.org/wiki/Stack-based_memory_allocation).

9) what does java use call by value or reference?

* Java does manipulate objects by reference, and all object variables are references. However, Java doesn't pass method arguments by reference; it passes them by value

10) what are wrapper [classes](http://www.glassdoor.com/Interview/MathWorks-Interview-Questions-E17117.htm)?

* Java is an object oriented programming language
* Wrapper classes are used instead of primitive types when an Object is expected.
* In **Collections** for example, an ArrayList may contain instances of any subclass of Object, but because primitive types are not Objects, they could not be contained in the ArrayList
* Am example of when wrappers are used would be in Collections, you can have an ArrayList<Integer>, but not an ArrayList<int> same with HashMaps etc. To get type safety we use generics and generics need objects not primitives

Java is an object-oriented language and as said everything in java is an object. But what about the primitives? They are sort of left out in the world of objects, that is, they cannot participate in the object activities, such as being returned from a method as an object, and being added to a Collection of objects, etc. . As a solution to this problem, Java allows you to include the primitives in the family of objects by using what are called **wrapper classes**.  
  
There is a wrapper class for every primitive date type in Java. This class encapsulates a single value for the primitive data type. For instance the wrapper class for int is Integer, for float is Float, and so on. Remember that the primitive name is simply the lowercase name of the wrapper except for char, which maps to Character, and int, which maps to Integer.  
  
**The wrapper classes in the Java API serve two primary purposes:**

* To provide a mechanism to “wrap” primitive values in an object so that the primitives can be included in activities reserved for objects, like as being added to Collections, or returned from a method with an object return value.
* To provide an assortment of utility functions for primitives. Most of these functions are related to various conversions: converting primitives to and from String objects, and converting primitives and String objects to and from different bases (or radix), such as binary, octal, and hexadecimal.

The wrapper object of a wrapper class can be created in one of two ways: by instantiating the wrapper class with the new operator or by invoking a static method on the wrapper class. We will explore this further in this article.

**Creating Wrapper Objects with the new Operator**

Before we can instantiate a wrapper class, we need to know its name and the arguments its constructor accepts. The name of the wrapper class corresponding to each primitive data type, along with the arguments its constructor accepts, is listed below:  
  
Primitive datatype-->Wrapper Class-->Constructor arguments

* boolean--> Boolean--> boolean or String
* byte--> Byte--> byte or String
* char--> Character--> char
* short--> Short--> short or String
* int-->Integer--> int or String
* long--> Long--> long or String
* float-->Float--> float double or String
* double-->Double--> double or String
* All the wrapper classes are declared final. That means you cannot derive a subclass from any of them.All the wrapper classes except Boolean and Character are subclasses of an abstract class called Number, whereas Boolean and Character are derived directly from the Object class.All of the wrapper classes except Character provide two constructors: one that takes a primitive of the type being constructed, and one that takes a String representation of the type being constructed—for example,

11) what is a abstract class?

* **Java Abstract classes** are used to declare common characteristics of subclasses. An abstract class cannot be instantiated. It can only be used as a superclass for other classes that extend the abstract class. Abstract classes are declared with the **abstract** keyword. Abstract classes are used to provide a template or design for concrete subclasses down the inheritance tree

12) what is a final keyword do?

* In the [Java programming language](http://en.wikipedia.org/wiki/Java_(programming_language)), the **final** [keyword](http://en.wikipedia.org/wiki/Keyword_(computing)) is used in several different contexts to define an entity which cannot later be changed
* A **final**[**class**](http://en.wikipedia.org/wiki/Class_(computer_science)) cannot be subclassed. This is done for reasons of security and efficiency

**final关键字的基本含义是: 这个数据/方法/类不能被改变了。**

* **final基本类型的数据: 定值 (constant value)，只能赋值一次，不能再被修改。**
* **final方法: 该方法不能被覆盖。private的方法默认为final的方法。**
* **final类: 该类不能被继承。**

### What is the difference between a.Equals(b) and a == b?

“==” compares reference – returns true if and only if both references point to the SAME object while   
"Equals" method compares object by VALUE and it will return true if the references refers object which are equivalent 

**Use of final, finally and finalize**

**Final:-** It is used in the following three cases:

* 1. If the final keyword is attached to a variable then the variable becomes constant i.e. its value cannot be changed in the program.
* 2. If a method is marked as final then the method cannot be overridden by any other method.
* 3. If a class is marked as final then this class cannot be inherited by any other class.

**Finally:-** If an exception is thrown in try block then the control directly passes to the catch block without executing the lines of code written in the remainder section of the try block. In case of an exception we may need to clean up some objects that we created. If we do the clean-up in try block, they may not be executed in case of an exception. Thus finally block is used which contains the code for clean-up and is always executed after the try ...catch block.  
  
**Finalize:-** It is a method present in a class which is called before any of its object is reclaimed by the garbage collector. Finalize() method is used for performing code clean-up before the object is reclaimed by the garbage collector.

* method **overloading** would be an example of static polymorphism
* whereas **overriding** would be an example of dynamic polymorphism

Signal Processing  
1) Aliasing?  
2) Z transform?

3) Fourier transform?  
4) Convolution?  
5) difference between convolution and correlation?  
6) what are anti- aliasing filters?  
7) impulse response?  
8) what is a iir and fir?  
9) is this iir or fir? hey gave me 4 sums on the same  
10) what are causal systems?  
11) what is a BIBO system? bounded system?  
12) gave me a question and asked me whether it is a stable/unstable? why?  
13) in a graph? tell me where does a stable system lies and unstable lies?  
  
Matlab  
1) What is a.\*b and a\*b?  
2) Script vs function  
3) workspace of a script vs workspace of a function?

* Scripts are m-files containing MATLAB codes that can be executed directly in the command window. MATLAB ``functions'' are another type of m-file. The biggest difference between scripts and functions is that functions have input and output parameters. Script files can only operate on the variables that are hard-coded into their m-file. As you can see, functions much more flexible. They are therefore more suitable for general purpose tasks that will be applied to different data
* Scripts are useful for tasks that don't change. They are also a way to document a specific sequence of actions, say a function call with special parameter values, that may be hard to remember
* There are more subtle differences between scripts and functions. A script can be thought of as a keyboard macro: when you type the name of the script, all of the commands contained in it are executed just as if you had typed these commands into the command window. Thus, all variables created in the script are added to the workspace for the current session

4) plot(x,y,'o'); what does that mean?

* Using plot(x,y, 'o') can draw a small circle in (x,y) point
* PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix, then the vector is plotted versus the rows or columns of the matrix, whichever line up

5) how to suppress comments

* Use % （persent sign）

# 6) how to plot position vs time and position vs distance on same graph?( in one figure

?)

* Use hold on and hold off when plotting
* Plot(p, t, ‘\*’, p, d, ‘o’)

7) how do you concatenate strings?

* strcat(A,B) horcar(a,b), vertcat(a,b)

8) how do you find the inverse of a matrix?

* inv
* A\I
* pinv and B\I (I is always the dimension of the number of the rows as B)

9) gave a problem and asked can you put 2 x axis or something of that sort? wasn't clear about the question.

* ??????????

10) how to find the number of elements in matlab? "numel" function

* numel(A) for matrix and vectors, length(a) for vectors
* atoi: convert string to integer

C/C++

-a linked list

* In [computer science](http://en.wikipedia.org/wiki/Computer_science), a **linked list** is a [data structure](http://en.wikipedia.org/wiki/Data_structure) consisting of a group of [nodes](http://en.wikipedia.org/wiki/Node_(computer_science)) which together represent a sequence. Under the simplest form, each node is composed of a datum and a [reference](http://en.wikipedia.org/wiki/Reference_(computer_science)) (in other words, a *link*) to the next node in the sequence;
* This structure allows for efficient insertion or removal of elements from any position in the sequence.

-advantages of linked list over array

* It is much convenient to insert or delete an element at a specific point of a linked list in terms of operation time. It only costs a constant time operation. While for array, it will require moving half of the elements on average, and all the elements in the worst case.
* Insertion or deletion of an element at a specific point of a list, is a constant-time operation, whereas insertion in a dynamic array at random locations will require moving half of the elements on average, and all the elements in the worst case
* Moreover, arbitrarily many elements may be inserted into a linked list, limited only by the total memory available; while a dynamic array will eventually fill up its underlying array data structure and will have to reallocate
* dynamic arrays (as well as fixed-size [array data structures](http://en.wikipedia.org/wiki/Array_data_structure)) allow constant-time [random access](http://en.wikipedia.org/wiki/Random_access), while linked lists allow only [sequential access](http://en.wikipedia.org/wiki/Sequential_access) to elements. Singly linked lists, in fact, can only be traversed in one direction
* disadvantage of linked lists is the extra storage needed for references

-method overriding

* **Method overriding**, in [object oriented programming](http://en.wikipedia.org/wiki/Object_oriented_programming), is a language feature that allows child class to provide a specific implementation of a [method](http://en.wikipedia.org/wiki/Method_(computer_science)) that is already provided by its parent classes.
* The implementation in the child class overrides (replaces) the implementation in the parent classes by providing a method that has same name, same [parameters](http://en.wikipedia.org/wiki/Parameter_(computer_science)) or signature, and same return type as the method in the parent class.[[1]](http://en.wikipedia.org/wiki/Method_overriding#cite_note-flanagan-1)
* The version of a method that is executed will be determined by the [object](http://en.wikipedia.org/wiki/Object_(computer_science)) that is used to invoke it

-function overloading

* **Function overloading** or **method overloading** is a feature found in various [programming languages](http://en.wikipedia.org/wiki/Programming_language) such as [Ada](http://en.wikipedia.org/wiki/Ada_(programming_language)), [C++](http://en.wikipedia.org/wiki/C%2B%2B), [C#](http://en.wikipedia.org/wiki/C_Sharp_(programming_language)), [D](http://en.wikipedia.org/wiki/D_(programming_language)), and [Java](http://en.wikipedia.org/wiki/Java_(programming_language)), that allows creating several [methods](http://en.wikipedia.org/wiki/Subprogram) with the same name which differ from each other in the type of the input and the output of the function. It is simply defined as the ability of one function to perform different tasks.

## Rules in function overloading

* The overloaded function must differ either by the [arity](http://en.wikipedia.org/wiki/Arity" \o "Arity) or data types.
* The same function name is used for various instances of function call.

# Scope resolution operator

namespace A

{ //

int i; //scope of A

} //

A::i = 4; //scope operator refers to the integer i declared under the namespace A

-a global variable and why is it considered a bad programming practice to use a global variable

* They clutter up the global namespace and are slower to look up than local variables.
* First of all, having many global variables is always a bad thing because it's easy to forget you declared a variable somewhere and accidentally re-declare it somewhere else
* a global variable can potentially be modified from anywhere (unless they reside in[protected memory](http://en.wikipedia.org/wiki/Protected_memory) or are otherwise rendered read-only), and any part of the program may depend on it

-a destructor

* In [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming), a **destructor** (sometimes shortened to **dtor**) is a [method](http://en.wikipedia.org/wiki/Method_(computer_science)) which is automatically invoked when the [object](http://en.wikipedia.org/wiki/Object_(computer_science)) is destroyed
* Its main purpose is to free the [resources](http://en.wikipedia.org/wiki/Resource_(computer_science)) (memory allocations, open files or sockets, database connections, resource locks, etc.) which were acquired by the object along its [life cycle](http://en.wikipedia.org/wiki/Destructor_(computer_programming)) and/or deregister from other entities which may keep references to it
* The destructor has the same name as the class, but with a [tilde](http://en.wikipedia.org/wiki/Tilde) (~) in front of it. If the object was created as an [automatic variable](http://en.wikipedia.org/wiki/Automatic_variable), its destructor is automatically called when it goes out of [scope](http://en.wikipedia.org/wiki/Scope_(programming)). If the object was created with a [new](http://en.wikipedia.org/wiki/New_(C%2B%2B)) expression, then its destructor is called when the [delete](http://en.wikipedia.org/wiki/Delete_(C%2B%2B)) operator is applied to a pointer to the object. Usually that operation occurs within another destructor, typically the destructor of a [smart pointer](http://en.wikipedia.org/wiki/Smart_pointer) object

-a constructor

* **constructor** in a [class](http://en.wikipedia.org/wiki/Class_(computer_science)) is a special type of [subroutine](http://en.wikipedia.org/wiki/Subroutine) called to [create an object](http://en.wikipedia.org/wiki/Object_lifetime#Creating_objects). It prepares the new object for use

-a default constructor

* In computer [programming languages](http://en.wikipedia.org/wiki/Programming_languages) the term “**default constructor**” refers to a [constructor](http://en.wikipedia.org/wiki/Constructor_(computer_science)) that is automatically generated in the absence of explicit constructors
* In [C++](http://en.wikipedia.org/wiki/C%2B%2B), the standard describes the default constructor for a class as a [constructor](http://en.wikipedia.org/wiki/Constructor_(computer_science)) that can be called with no arguments (this includes a constructor whose parameters all have default arguments)
* In C++, default constructors are significant because they are automatically invoked in certain circumstances:
* When an object value is declared with no argument list, e.g. MyClass x;; or allocated dynamically with no argument list, e.g. new MyClass or new MyClass(); the default constructor is used to initialize the object
* When an array of objects is declared, e.g. MyClass x[10];; or allocated dynamically, e.g. new MyClass [10]; the default constructor is used to initialize all the elements
* When a derived class constructor does not explicitly call the base class constructor in its initializer list, the default constructor for the base class is called
* When a class constructor does not explicitly call the constructor of one of its object-valued fields in its initializer list, the default constructor for the field's class is called
* In the standard library, certain containers "fill in" values using the default constructor when the value is not given explicitly, e.g. vector<MyClass>(10);initializes the vector with 10 elements, which are filled with the default-constructed value of our type.
* In the above circumstances, it is an error if the class does not have a default constructor.

-a pointer

* In [computer science](http://en.wikipedia.org/wiki/Computer_science), a **pointer** is a  [data type](http://en.wikipedia.org/wiki/Data_type) whose value refers directly to (or "**points** to") another value stored elsewhere in the [computer memory](http://en.wikipedia.org/wiki/Computer_memory) using its [address](http://en.wikipedia.org/wiki/Memory_address)
* A pointer *references* a location in memory, and obtaining the value stored at that location is known as *dereferencing*the pointer
* Reference operator (&)
* Dereference operator (\*)

-the difference between a struct and a class

* If you don't specify public: or private:, members of a struct are public by default; members of a class are private by default
* members of a class are private by default, whereas members of a struct are public by default.
* Inheritance between classes is also private by default, and inheritance between structs is public by default
* The members and base classes of a struct are public by default, while in class, they default to private. Note: you should make your base classes explicitly public, private, or protected, rather than relying on the defaults.
* struct and class are otherwise functionally equivalent

-the difference between a directed graph and a tree

* a **tree** is an [undirected graph](http://en.wikipedia.org/wiki/Undirected_graph) in which any two [vertices](http://en.wikipedia.org/wiki/Vertex_(graph_theory)) are connected by *exactly one* [simple path](http://en.wikipedia.org/wiki/Path_(graph_theory)). In other words, any [connected](http://en.wikipedia.org/wiki/Connectedness) [graph](http://en.wikipedia.org/wiki/Tree_(graph_theory)) without [simple cycles](http://en.wikipedia.org/wiki/Cycle_(graph_theory)) is a tree
* A Tree is just a restricted form of a Graph.
* Trees have direction (parent / child relationships) and don't contain cycles. They fit with in the category of Directed Acyclic Graphs (or a DAG). So Trees are DAGs with the restriction that a child can only have one parent.
* One thing I think important to point out, Trees aren't a recursive data structure. They can be implemented as a recursive data structure because of the above restrictions. But any DAG implementation, which are generally not recursive, can also be used. My preferred Tree implementation is a centralized map representation and is non recursive
* Graphs are generally search breath first or depth first. The same applies to Tree
* Trees are obvious: they're recursive data structures consisting of nodes with children

(some more inheritance & OOP concepts)

11) what is a abstract class?

* An abstract class is a class that cannot be instantiated and is usually implemented as a class that has one or more **pure virtual** functions
* In general an abstract class is used to define an implementation and is intended to be inherited from by concrete classes
* If we wish to create a concrete class (a class that can be instantiated) from an abstract class we must declare and **define** a matching member function for each abstract member function of the **base class**
* An **abstract class** is a class that is designed to be specifically used as a **base class**. An abstract class contains at least one pure virtual function.
* You declare a pure virtual function by using a pure specifier (= 0) in the declaration of a virtual member function in the class declaration

-a virtual function

* In [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming), a **virtual function**  is a [function](http://en.wikipedia.org/wiki/Function_(computer_science)) whose behavior can be [overridden](http://en.wikipedia.org/wiki/Method_overriding_(programming)) within an inheriting class by a function with the same [signature](http://en.wikipedia.org/wiki/Method_signature). This concept is a very important part of the[polymorphism](http://en.wikipedia.org/wiki/Polymorphism_(computer_science)) portion of [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming) (OOP).

12) pure virtual function

* A pure virtual function is one which **must be overridden** by any concrete derived class. Its declaration is with the **syntax " = 0"** in the [member](http://en.wikibooks.org/wiki/C%2B%2B_Programming/Classes/Abstract_Classes) function's declaration.
* **virtual function** is a [function](http://en.wikipedia.org/wiki/Function_(computer_science))  whose behavior can be [overridden](http://en.wikipedia.org/wiki/Method_overriding_(programming)) within an inheriting class by a function with the same [signature](http://en.wikipedia.org/wiki/Method_signature). This concept is a very important part of the[polymorphism](http://en.wikipedia.org/wiki/Polymorphism_(computer_science)) portion of [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming) (OOP)

13) diff between malloc and new

* new will not only allocate you memory for objects but will also call the constructor on the objects created.
* malloc will just allocate you a block of uninitialized memory of the given size, with no guarantee of the contents. The allocated memory has to be released with free.
* new and delete are C++ specific features. They didn't exist in C. malloc is the old [school](http://stackoverflow.com/questions/807939/what-is-the-difference-between-new-and-malloc-and-calloc-in-c) C way to do things. Most of the time, you won't need to use it in C++.

**new/delete + new[]/delete[]:**

* new/delete is the C++ way to allocate memory and deallocate memory from the heap.
* new[] and delete[] is the c++ way to allocate arrays of contiguous memory.
* Should be used because **it is more type safe than malloc**
* Should be used because **it calls the constructor/destructor**
* Cannot be used in a realloc way, but can use placement new to re-use the same buffer of data
* **Data** cannot be allocated with new and freed with free, nor delete[]

**malloc/free + family:**

* malloc/free/family is the C way to allocate and free memory from the heap.
* calloc is the same as malloc but also initializes the memory
* Should be used if you may need to reallocate the memory
* Data cannot be allocated with malloc and freed with delete nor delete[]

Expand A ^ ( B v C )

* This is not De Morgan’s law, A intersects the union of B and C equals the union of A and B intersects the union of A and C

Your friend and you are playing a coin toss game. Your friend toss the coin and he tells you what the result is. If the coin toss is head, he tells the truth. If it is tail, he tells the opposite.  
What's the probability that the coin toss is head if he tells you it's a head.

* 1/2

2. Pointers?  
3. Null Pointer?

* We said that the value of a pointer variable is a pointer to some other variable. There is one other value a pointer may have: it may be set to a null pointer. A null pointer is a special pointer value that is known not to point anywhere. What this means that no other valid pointer, to any other variable or array cell or anything else, will ever compare equal to a null pointer
* Dereferencing the NULL pointer typically results in an attempted read or write from memory that is not mapped

4. Difference between & and \* ?

* & is to get the address of variable in the memory, while \* is to get the value stored in the address of memory
* The address that locates a variable within memory is what we call a *reference* to that variable. This reference to a variable can be obtained by preceding the identifier of a variable with an ampersand sign (&), known as reference operator

* The single ampersand & is the logical AND operator. The double ampersand && is again a logical AND operator that employs short-circuiting behaviour. Short-circuiting just means the second operand (right hand side) is evaluated only when the result is not fully determined by the first operand (left hand side)
* A & B (A and B are evaluated)
* A && B (B is only evaluated if A is true)

6. how would you add path in command line?

* cd('D:\userdata\Desktop\CS 648 Computer Security')
* addpath

7. How you save a variable in a mat file?

* Save + the name of the variable in the workplace, will be saved as something.malt

8. How you save some specific variables in a mat file?

* Save + the name of the variable in the workplace, will be saved as something.malt
* save(*filename*) stores all variables from the current workspace in a MATLAB formatted binary file (MAT-file) called *filename*.
* save(*filename*, *variables*) stores only the specified variables.

9. How you open these data?

* S = load(filename) loads the variables from a MAT-file into a structure array, or data from an ASCII file into a double-precision array.
* S = load(filename, variables) loads the specified variables from a MAT-file.

10. How can you see the type of a variable?

* Whos
* Class(a)

11.How can you do the inverse of a matrix? (OK)

12. How you do the transpose of a matrix?

* b = a.' computes the non-conjugate transpose of matrix a and returns the result in b.
* b = transpose(a) is called for the syntax a.' when a is an object.
* b = a' computes the complex conjugate transpose of matrix a and returns the result in b.
* b = ctranspose(a) is called for the syntax a' (complex conjugate transpose) when a is an object.
* a+bi, in this expression, *a* is the real part and *b* is the imaginary part of the complex number

13. What A.\*B does?

|  |  |
| --- | --- |
| .\* | Array multiplication. A.\*B is the element-by-element product of the arrays A and B. A and B must have the same size,  unless one of them is a scalar. |
| .^ | Array power. A.^B is the matrix with elements A(i,j) to the B(i,j) power. A and B must have the same size, unless one of  them is a scalar. |
| ' | Matrix transpose. A' is the linear algebraic transpose of A. For complex matrices, this is the complex conjugate transpose. |
| .' | Array transpose. A.' is the array transpose of A. For complex matrices, this does not involve conjugation. |

14. What A\*B does?

**Interview Details –** Round I: General [background check](http://www.glassdoor.com/Interview/MathWorks-Applications-Support-Engineer-Interview-Questions-EI_IE17117.0,9_KO10,39.htm). 20 mins. call by HR person. Asked to select a section among - Controls, Signal Processing, Embedded Systems, Computer Science.  
  
Round II: Rapid fire question-answer session for about 40 - 45 mins. with no instant feedback as such. Told will get a call in two weeks, if interested further. Got an email in three days for another round (on-site).  
  
Round III: Pleasant stay at the Natick Courtyard. Interview on the next day for half a day. Technical round:  
Math - Graph plots, Decay equation, Integration, Vector projection, etc.

Decay equation:

Vector projection:

Programming logic - Recursive, Fibonacci, Sorting.  
C++ - Inheritance, Pointers.  
Signal Processing - Convolution, Sampling theory, IIR and FIR filters.  
Matlab - Wh questions, matrix configurations (eg., display A(:) for 3x3 matrix).  
Hiring Manager:  
Told her story first. Then asked general background/life story questions.  
Lunch:  
Informal lunch meeting with two ASEs.  
HR round:  
More background questions and behavioral ones. Explained the company details, hiring policy, benefits, etc.

OOP:

OOP is a programming paradigm that treats concepts as objects that have data fields, and associated procedures known as methods. Objects, which are usually [instances](http://en.wikipedia.org/wiki/Instance_(computer_science)) of [classes](http://en.wikipedia.org/wiki/Class_(computer_science)), are used to interact with one another to design applications and computer programs

* **Object-oriented programming** (**OOP**) is a [programming paradigm](http://en.wikipedia.org/wiki/Programming_paradigm)  that represents concepts as "[objects](http://en.wikipedia.org/wiki/Object_(computer_science))" that have [data fields](http://en.wikipedia.org/wiki/Field_(computer_science)) (attributes that describe the object) and associated procedures known as [methods](http://en.wikipedia.org/wiki/Method_(computer_science)). Objects, which are usually [instances](http://en.wikipedia.org/wiki/Instance_(computer_science)) of [classes](http://en.wikipedia.org/wiki/Class_(computer_science)), are used to interact with one another to design applications and computer programs
* An object-oriented program may be viewed as a collection of interacting *objects*, as opposed to the conventional model, in which a program is seen as a list of tasks ([subroutines](http://en.wikipedia.org/wiki/Subroutine)) to perform. In OOP, each object is capable of receiving messages, processing data, and sending messages to other objects
* In contrast, the object-oriented approach encourages the programmer to place data where it is not directly accessible by the rest of the program. Instead, the data is accessed by calling specially written functions, commonly called [methods](http://en.wikipedia.org/wiki/Method_(computer_science)), which are bundled in with the data

Inheritance:

* "Subclasses" are more specialized versions of a class, which inherit attributes and behaviors from their parent classes, and can introduce their own
* In [object-oriented programming](http://en.wikipedia.org/wiki/Object-oriented_programming) (OOP), **inheritance** is a way to establish [Is-a](http://en.wikipedia.org/wiki/Is-a) relationship between objects.
* Re-usability of code is achieved through composition ([Composition over inheritance](http://en.wikipedia.org/wiki/Composition_over_inheritance)). In *classical inheritance* where objects are defined by [classes](http://en.wikipedia.org/wiki/Class_(computer_programming)), classes can inherit attributes and behavior from pre-existing classes called [base classes](http://en.wikipedia.org/wiki/Base_class), *superclasses*, or *parent classes*

2) Benefits of class

* When you move a lot of the logic into a custom class, you can not only make the code easier to reuse, but you can also "hide" some of the methods and properties from other parts of the ActionScript code. This helps you prevent people from accessing sensitive information or changing data that shouldn't be changed
* When you use a class, you can also **extend** existing classes and **add new functionality** or **modify existing functionality**. For example, if you create three very similar classes, you can write a base class and then write two other classes that extend the base class. These two classes can add additional methods and properties
* **Another benefit of using classes is code reusability**

Advantages of putting everything in header files:

* Less redundancy (which leads to easier changes, easier refactoring, etc.)
* May give compiler/linker better opportunities for optimization
* Often easier to incorporate into an existing project

Disadvantages of putting everything in header files:

* Longer compile/link cycles
* Loss of separation of interface and implementation
* Sometimes leads to hard-to-resolve circular dependencies
* Lots of inlining could increase executable size
* Prevents binary compatibility of shared libraries/DLLs
* Pisses off many co-workers who prefer the traditional ways of using C++

Pigeon hole principle, max flow min cut theoram, finite state machine,  
In the nutshell, all were basic programming languages, no algorithm / data structure questions was asked.

**Interview Question –** difference between null and void pointer in C++.   [**Answer Question**](http://www.glassdoor.com/Interview/difference-between-null-and-void-pointer-in-C-QTN_420558.htm)

The interview will take approximately 40 minutes and will cover:  
  
Basic math - calculus, differential equations, and linear algebra;  
Programming concepts -- syntax and how languages work;  
One programming language – C (recommended), C++, or Java - if C++ is selected, you must understand syntax and structure, compilation and OOP concepts;  
MATLAB -- basic MATLAB syntax, data and programming concepts;  
Signals Processing (concepts such as aliasing, convolution, and transforms);

The interview was in 3 sections.  
1. Basic Programming Sections.  
2. Programming language specific questions (C++, Java).  
3. Basic Math.  
  
Questions were pretty generic  
like why is java platform independent.  
In which cases would you use pointers  
pointer safety.

# Pigeonhole principle

* In [mathematics](http://en.wikipedia.org/wiki/Pigeonhole_principle), the **pigeonhole principle** states that if *n* items are put into *m* [pigeonholes](http://en.wiktionary.org/wiki/pigeonhole) with *n* > *m*, then at least one pigeonhole must contain more than one item

# Max-flow min-cut theorem

* **in a flow network, the value of the max flow is equal to the capacity of the min cut**
* the **generalized max-flow min-cut theorem** states that the maximum value of an s-t flow is equal to the minimum capacity of an s-t cut in the new sense
* **the bottleneck determines the maximum**

# finite state machine

* A **finite-state machine** is conceived as an [abstract machine](http://en.wikipedia.org/wiki/Abstract_machine) that can be in one of a finite number of [*states*](http://en.wikipedia.org/wiki/State_(computer_science)). The machine is in only one state at a time
* It can change from one state to another when initiated by a triggering event or condition; this is called a *transition*. A particular FSM is defined by a list of its states, and the triggering condition for each transition.

# four color map theorem

* states that, given any separation of a plane into [contiguous](http://en.wikipedia.org/wiki/Contiguity#Geography) regions, producing a figure called a *map*, no more than four colors are required to color the regions of the map so that no two adjacent regions have the same color

Turing Machine

* A Turing machine is a problem decider, that given an input w and halts in either q\_accept or q\_reject start.
* a polynomial time bounded Turing Machine is a Turing maching which given a input w with length of n will halt within p(n) steps, where p(n) is a polynomial in n.

In which cases would you use pointers

* When you want to construct some date structure, say linked list.

pointer safety

* Avoid pointers until you can't.
* The reason is that pointers makes things harder to follow/read, less safe and far more dangerous manipulations than any other constructs.
* So the rule of thumbs is to use pointers only if there is no other choice.
* For example, returning a pointer to an object is a valid option when the function can return nullptr in some cases and it is assumed it will. That said, a better option would be to use something similar to boost::optional.
* Another example is to use pointers to raw memory for specific memory manipulations. That should be hidden and localized in very narrow parts of the code, to help limit the dangerous part of the whole code base.

Main reasons for using pointers:

* control object lifetime;
* can't use references (e.g. you want to store something non-copyable in vector);
* you should pass pointer to some third party function;
* maybe some optimization reasons, but I'm not sure
* In [linear algebra](http://en.wikipedia.org/wiki/Linear_algebra), the **kernel** or **null space** (also **nullspace**) of a [matrix](http://en.wikipedia.org/wiki/Matrix_(mathematics)) **A** is the [set](http://en.wikipedia.org/wiki/Set_(mathematics)) of all vectors **x** for which **Ax** = **0**. The kernel of a matrix with [real](http://en.wikipedia.org/wiki/Real_number) coefficients and*n* columns is a [linear subspace](http://en.wikipedia.org/wiki/Euclidean_subspace) of *n*-dimensional [Euclidean space](http://en.wikipedia.org/wiki/Euclidean_space).[[1]](http://en.wikipedia.org/wiki/Kernel_(matrix)#cite_note-textbooks-1) The [dimension](http://en.wikipedia.org/wiki/Dimension_(vector_space)) of the null space of **A** is called the **nullity** of **A**
* In [mathematics](http://en.wikipedia.org/wiki/Mathematics), the **rank–nullity theorem** of [linear algebra](http://en.wikipedia.org/wiki/Linear_algebra), in its simplest form, states that the [rank](http://en.wikipedia.org/wiki/Rank_(matrix_theory)) and the [nullity](http://en.wikipedia.org/wiki/Kernel_(matrix)) of a matrix add up to the number of columns of the matrix. Specifically, if *A* is an *m*-by-*n* matrix (with *m* rows and *n* columns) over some [field](http://en.wikipedia.org/wiki/Field_(mathematics)), then
* rank *A* + nullity *A* = *n*.[[1]](http://en.wikipedia.org/wiki/Rank%E2%80%93nullity_theorem#cite_note-1)
* Solve [A 0] to find the null space

3. C:  
1) What is pointer?  
2) What is typedef used for?

* The purpose of typedef is to assign alternative names to existing [types](http://en.wikipedia.org/wiki/Type_system)
* typedef struct MyStruct{};

3) What is the difference between malloc and new?  
  
4. Matlab:  
1) How to merge two strings?

* strcat()

2) What is the difference between function and script?

* Scripts are m-files containing MATLAB statements. MATLAB ``functions'' are another type of m-file. The biggest difference between scripts and functions is that functions have input and output parameters. Script files can only operate on the variables that are hard-coded into their m-file. As you can see, functions much more flexible. They are therefore more suitable for general purpose tasks that will be applied to different data
* Scripts are useful for tasks that don't change. They are also a way to document a specific sequence of actions, say a function call with special parameter values, that may be hard to remember
* There are more subtle differences between scripts and functions. A script can be thought of as a keyboard macro: when you type the name of the script, all of the commands contained in it are executed just as if you had typed these commands into the command window. Thus, all variables created in the script are added to the workspace for the current session

3) How to access the variable hide in a function?

* let the function return that variable?

4) Deterministic and Non Deterministic

* The class of P is the set of problems which can be solved by a deterministic polynomial time bounded Turing machine
* The class of NP is the set of problems which can be solved by a non-deterministic polynomial time bounded Turing machine
* Polynomial time mapping reducible
* A problem is NP-C if and only if every problem in NP is polynomial time mapping reducible to this problem and L belongs to NP first.

Some questions showed up in my interview:

three cautions to use pointers: ???

multi-thread

#define y=20 and constd int x=10;

The name of convert double into int

* Type casting
* Typecasting is making a variable of one type, such as an int, act like another type, a char, for one single operation. To typecast something, simply put the type of variable you want the actual variable to act as inside parentheses in front of the actual variable. (char)a will make 'a' function as a char
* Standard conversions affect fundamental data types, and allow conversions such as the conversions between numerical types (short to int, int to float, double to int...), to or from bool, and some pointer conversions. Some of these conversions may imply a loss of precision, which the compiler can signal with a warning. This can be avoided with an explicit conversion
* Example: int a=5, b=50;
* Float c=a/b; // output c=0
* Float c=(float)a/b; //output c=0.1

Storage class in C

* C has a concept of '*Storage classes*' which are used to define the scope (visability) and life time of variables and/or functions.
* auto register static extern typedef
* **auto** is the default storage class for local variables
* **register** is used to define local variables that should be stored in a register instead of RAM. This means that the variable has a maximum size equal to the register size (usually one word) and cant have the unary '&' operator applied to it (as it does not have a memory location).
* Register should only be used for variables that require quick access - such as counters. It should also be noted that defining 'register' goes not mean that the variable will be stored in a register. It means that it MIGHT be stored in a register - depending on hardware and implimentation restrictions
* **static** is the default storage class for [global variables](http://www.lix.polytechnique.fr/~liberti/public/computing/prog/c/C/SYNTAX/glo_int_vars.html#global).
* **extern** defines a global variable that is visable to ALL object modules. When you use 'extern' the variable cannot be initalized as all it does is point the variable name at a storage location that has been previously defined
* **typedef** is used to define new data type names to make a program more readable to the programmer.

Union and structure in C

* The **union** [keyword](http://www.lix.polytechnique.fr/~liberti/public/computing/prog/c/C/glossary.html#keyword) allows several variables of different type and size to occupy the same storage location.
* **struct** is used to declare a new [data-type](http://www.lix.polytechnique.fr/~liberti/public/computing/prog/c/C/CONCEPT/data_types.html). Basically this means grouping variables together
* The difference between structure and union in c are: 1. union allocates the memory equal to the maximum memory required by the member of the union but structure allocates the memory equal to the total memory required by the members. 2. In union, one block is used by all the member of the union but in case of structure, each member have their own memory space
* With a union, all members share the same memory. With a struct, they do not share memory, so a different space in memory is allocated to each member of the struct

Give the two enficient conditions to make a function f linear

* F(x+y)=f(x)+f(y)
* F(ax)=af(x)

Data ?

What is hash function?

Hash table, BST, array, linked list

Caches

* 1. Pronounced as Cash(like the money). **Cache** is a high-speed access area that can be either a reserved section of main [memory](http://www.computerhope.com/jargon/m/memory.htm) or a storage device. The two main cache types are **memory cache** and **disk cache**. Memory cache is a portion on memory of high-speed static [RAM](http://www.computerhope.com/jargon/r/ram.htm) ([SRAM](http://www.computerhope.com/jargon/s/sram.htm)) and is effective because most programs access the same data or instructions over-and-over. By keeping as much of this information as possible in SRAM, the computer avoids accessing the slower [DRAM](http://www.computerhope.com/jargon/d/dram.htm). Most computers today come with [L3 cache](http://www.computerhope.com/jargon/l/l3.htm) or [L2 cache](http://www.computerhope.com/jargon/l/l2.htm), while older computers included only [L1 cache](http://www.computerhope.com/jargon/l/l1.htm).
* 2. Like memory **caching**, **disk caching** is used to access commonly accessed data. However, instead of using high-speed SRAM, a disk cache uses [conventional](http://www.computerhope.com/jargon/c/conventi.htm) main memory. The most recently accessed data from a disk is stored in a memory buffer. When a program needs to access data from the disk, it first checks the disk cache to see if the data is there. Disk caching can dramatically improve the performance of applications because accessing a byte of data in RAM can be thousands of times faster than accessing a byte on a hard drive.
* 3. Another **cache** is known as "**Internet browser cache**" also known as "Temporary Internet Files" in Internet Explorer. Internet cache is used to help improve how fast data is opened while browsing the Internet. In most cases, each time a [web page](http://www.computerhope.com/jargon/w/webpage.htm) is opened, it is sent to your browser's temporary cache on your [hard drive](http://www.computerhope.com/jargon/h/harddriv.htm). If that page is accessed again and has not been modified, the browser will open the page from your cache instead of [downloading](http://www.computerhope.com/jargon/d/download.htm) the page again. This saves users a lot of time, especially if that the user is using a [modem](http://www.computerhope.com/jargon/m/modem.htm), and can also help save the web page owner on [bandwidth](http://www.computerhope.com/jargon/b/bandwidt.htm).

Mutex?

* Short for ***mu***tual***ex***clusion object. In computer programming, a mutex is a program [object](http://www.webopedia.com/TERM/O/object.html) that allows multiple program [threads](http://www.webopedia.com/TERM/T/thread.html) to **share the same resource**, such as file access, but not simultaneously. When a program is started, a mutex is created with a unique name. After this stage, any thread that needs the resource must lock the mutex from other threads while it is using the resource. The mutex is set to unlock when the data is no longer needed or the routine is finished
* The point of a mutex is to synchronize two threads. When you have two threads attempting to access a single resource, the general pattern is to have the first block of code attempting access to set the mutex before entering the code. When the second code block attempts access, it sees the mutex is set and waits until the first block of code is complete (and un-sets the mutex), then continues

Local scope v.s. global scope

* C++ names can be used only in certain regions of a program. This area is called the "scope" of the name.
* Scope determines the "lifetime" of a name that does not denote an object of static extent. Scope also determines the visibility of a name, when class constructors and destructors are called, and when variables local to the scope are initialized
* **Local scope**A name declared within a block is accessible only within that block and blocks enclosed by it, and only after the point of declaration. The names of formal arguments to a function in the scope of the outermost block of the function have local scope, as if they had been declared inside the block enclosing the function body
* **Function scope**Labels are the only names that have function scope. They can be used anywhere within a function, but are not accessible outside that function
* **File scope**Any name declared outside all blocks or classes has file scope. It is accessible anywhere in the translation unit after its declaration. Names with file scope that do not declare static objects are often called global names
* In C++, file scope is also known as namespace scope
* **Class scope**Names of class members have class scope. Class member functions can be accessed only by using the member-selection operators (**.**or **–>**) or pointer-to-member operators (**.\*** or **–>\***) on an object or pointer to an object of that class; nonstatic class member data is considered local to the object of that class. Consider the following class declaration
* **Prototype scope**Names declared in a function prototype are visible only until the end of the prototype. The following prototype declares three names (strDestination, numberOfElements, and strSource); these names go out of scope at the end of the prototype

#### Global scope

* The default scope is defined as **global scope**, this is commonly used to define and use global variables or other global constructs (classes, structure, functions, etc...), this makes them valid and visible to the compiler at all times
* increment the value of void pointer’s variable? Pointer arithmetic is not allowed on void\* pointers.

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

* The difference between "" and <> isn't much. Both search for the header in implementation-defined places1, 2. The difference is that if that search fails for "", the search happens as if it was using <>.
* This is because of how the include syntax is defined.
* #include <cstdio> means that the compiler should include the standard library cstdio
* #include "cstdio" means the compiler should try to find the file "cstdio", looking primarily in the current directory and using the location of the standard libraries as a fallback
* precompiler will start lookup from the directory of translation unit, and then move to predefined "include" directories

* They're exceptions that occur when you try to use a reference that points to no location in memory (null) as though it were referencing an object. Calling a method on a null reference or trying to access a field of a null reference will trigger a NPE
* A null pointer exception is caused when you dereference a variable that is pointing to null

Conflict can be pretty much inevitable when you work with others. People have different viewpoints and under the right set of circumstances, those differences escalate to conflict. How you handle that conflict determines whether it works to the team's advantage or contributes to its demise.

You can choose to ignore it, complain about it, blame someone for it, or try to deal with it through hints and suggestions; or you can be direct, clarify what is going on, and attempt to reach a resolution through common techniques like negotiation or compromise. It's clear that conflict has to be dealt with, but the question is how: It has to be dealt with constructively and with a plan, otherwise it's too easy to get pulled into the argument and create an even larger mess.

Conflict isn't necessarily a bad thing, though. Healthy and constructive conflict is a component of high functioning teams. Conflict arises from differences between people; the same differences that often make diverse teams more effective than those made up of people with similar experience. When people with varying viewpoints, experiences, skills, and opinions are tasked with a project or challenge, the combined effort can far surpass what any group of similar individual could achieve. Team members must be open to these differences and not let them rise into full-blown disputes.

Understanding and appreciating the various viewpoints involved in conflict are key factors in its resolution. These are key skills for all team members to develop. The important thing is to maintain a healthy balance of constructive difference of opinion, and avoid negative conflict that's destructive and disruptive.

Getting to, and maintaining, that balance requires well-developed team skills, particularly the ability to resolve conflict when it does happens, and the ability to keep it healthy and avoid conflict in the day-to-day course of team working. Let's look at conflict resolution first, then at preventing it.

## Resolving Conflict

When a team oversteps the mark of healthy difference of opinion, resolving conflict requires respect and patience. The human experience of conflict involves our emotions, perceptions, and actions; we experience it on all three levels, and we need to address all three levels to resolve it. We must replace the negative experiences with positive ones.

The three-stage process below is a form of mediation process, which helps team members to do this:

### Step 1: Prepare for Resolution

* **Acknowledge the conflict** – The conflict has to be acknowledged before it can be managed and resolved. The tendency is for people to ignore the first signs of conflict, perhaps as it seems trivial, or is difficult to differentiate from the normal, healthy debate that teams can thrive on. If you are concerned about the conflict in your team, discuss it with other members. Once the team recognizes the issue, it can start the process of resolution.
* **Discuss the impact** – As a team, discuss the impact the conflict is having on team dynamics and performance.
* **Agree to a cooperative process** – Everyone involved must agree to cooperate in to resolve the conflict. This means putting the team first, and may involve setting aside your opinion or ideas for the time being. If someone wants to win more than he or she wants to resolve the conflict, you may find yourself at a stalemate.
* **Agree to communicate** – The most important thing throughout the resolution process is for everyone to keep communications open. The people involved need to talk about the issue and discuss their strong feelings. [**Active listening**](http://www.mindtools.com/CommSkll/ActiveListening.htm) is essential here because to move on you need to really understand where the other person is coming from.

### Step 2: Understand the Situation

Once the team is ready to resolve the conflict, the next stage is to understand the situation, and each team member's point of view. Take time to make sure that each person's position is heard and understood. Remember that strong emotions are at work here so you have to get through the emotion and reveal the true nature of the conflict.

* **Clarify positions** – Whatever the conflict or disagreement, it's important to clarify people's positions. Whether there are obvious factions within the team who support a particular option, approach or idea, or each team member holds their own unique view, each position needs to be clearly identified and articulated by those involved.
* This step alone can go a long way to resolve the conflict, as it helps the team see the facts more objectively and with less emotion.

http://www.mindtools.com/images/box/top460grey.png

Sally and Tom believe the best way to market the new product is through a TV campaign. Mary and Beth are adamant that internet advertising is the way to go; whilst Josh supports a store-lead campaign.

http://www.mindtools.com/images/box/bottom460grey.png

* **List facts, assumptions and beliefs underlying each position** – What does each group or person believe? What do they value? What information are they using as a basis for these beliefs? What decision-making criteria and processes have they employed?

http://www.mindtools.com/images/box/top460grey.png

Sally and Tom believe that TV advertising is best because it has worked very well in the past. They are motivated by the saying, "If it ain't broke, don't fix it."

Mary and Beth are very tuned-in to the latest in technology and believe that to stay ahead in the market, the company has to continue to try new things. They seek challenges and find change exhilarating and motivating. Josh believes a store-lead campaign is the most cost-effective. He's cautious, and feels this is the best way to test the market at launch, before committing the marketing spend.

http://www.mindtools.com/images/box/bottom460grey.png

* **Analyze in smaller groups** – Break the team into smaller groups, separating people who are in alliance. In these smaller groups, analyze and dissect each position, and the associated facts, assumptions and beliefs.
* Which facts and assumptions are true? Which are the more important to the outcome? Is there additional, objective information that needs to be brought into the discussion to clarify points of uncertainly or contention? Is additional analysis or evaluation required?

http://www.mindtools.com/images/box/top440grey.png

**Tip:**  
Consider using formal evaluation and decision-making processes where appropriate. Techniques such as [**PMI**](http://www.mindtools.com/pages/article/newTED_05.htm), [**Force Field Analysis**](http://www.mindtools.com/pages/article/newTED_06.htm), [**Paired Comparison Analysis**](http://www.mindtools.com/pages/article/newTED_02.htm) and [**Cost/Benefit Analysis**](http://www.mindtools.com/pages/article/newTED_08.htm) are among those that could help.

If such techniques have not been used already, they may help make a much more objective decision or evaluation. Gain agreement within the team about which techniques to use, and how to go about the further analysis and evaluation.

http://www.mindtools.com/images/box/bottom440grey.png

* By considering the facts, assumptions, beliefs and decision making that lead to other people's positions, the group will gain a better understanding of those positions. Not only can this reveal new areas of agreement, it can also reveal new ideas and solutions that make the best of each position and perspective.
* Take care to remain open, rather than criticize or judge the perceptions and assumptions of other people. Listen to all solutions and ideas presented by the various sides of the conflict. Everyone needs to feel heard and acknowledged if a workable solution is to be reached.
* **Convene back as a team** – After the group dialogue, each side is likely to be much closer to reaching agreement. The process of uncovering facts and assumptions allows people to step away from their emotional attachments and see the issue more objectively. When you separate alliances, the fire of conflict can burn out quickly, and it is much easier to see the issue and facts laid bare.

### Step 3: Reach Agreement

Now that all parties understand the others' positions, the team must decide what decision or course of action to take. With the facts and assumptions considered, it's easier to see the best of action and reach agreement.

http://www.mindtools.com/images/box/top460grey.png

In our example, the team agrees that TV advertising is the best approach. It has had undeniably great results in the past and there is no data to show that will change. The message of the advertising will promote the website and direct consumers there. This meets Mary and Beth's concern about using the website for promotions: they assumed that TV advertising would disregard it.

http://www.mindtools.com/images/box/bottom460grey.png

If further analysis and evaluation is required, agree what needs to be done, by when and by whom, and so plan to reach agreement within a particular timescale. If appropriate, define which decision making and evaluation tools are to be employed.

If such additional work is required, the agreement at this stage is to the approach itself: Make sure the team is committed to work with the outcome of the proposed analysis and evaluation.

http://www.mindtools.com/images/box/top460grey.png

**Tip:**  
If the team is still not able to reach agreement, you may need to use a techniques like [**Win-Win Negotiation**](http://www.mindtools.com/CommSkll/NegotiationSkills.htm), [**Nominal Group Technique**](http://www.mindtools.com/pages/article/newTED_98.htm) or [**Multi-Voting**](http://www.mindtools.com/pages/article/newTMM_97.htm) to find a solution that everyone is happy to move the team ahead.

http://www.mindtools.com/images/box/bottom460grey.png

When conflict is resolved take time to celebrate and acknowledge the contributions everyone made toward reaching a solution. This can build team cohesion and confidence in their problem solving skills, and can help avert further conflict.

This three-step process can help solve team conflict efficiently and effectively. The basis of the approach is gaining understanding of the different perspectives and using that understanding to expand your own thoughts and beliefs about the issue.

## Preventing Conflict

As well as being able to handle conflict when it arises, teams need to develop ways of preventing conflict from becoming damaging. Team members can learn skills and behavior to help this. Here are some of the key ones to work on:

* Dealing with conflict immediately – avoid the temptation to ignore it.
* Being open – if people have issues, they need to be expressed immediately and not allowed to fester.
* Practicing clear communication – articulate thoughts and ideas clearly.
* Practicing active listening – paraphrasing, clarifying, questioning.
* Practicing identifying assumptions – asking yourself "why" on a regular basis.
* Not letting conflict get personal – stick to facts and issues, not personalities.
* Focusing on actionable solutions – don't belabor what can't be changed.
* Encouraging different points of view – insist on honest dialogue and expressing feelings.
* Not looking for blame – encourage ownership of the problem and solution.
* Demonstrating respect – if the situation escalates, take a break and wait for emotions to subside.
* Keeping team issues within the team – talking outside allows conflict to build and fester, without being dealt with directly.

To explore the process of conflict resolution in more depth, take our Bite-Sized Training session on [**Dealing with Conflict**](http://www.mindtools.com/community/Bite-SizedTraining/DealingWithConflict.php).

http://www.mindtools.com/images/box/top460grey.png

## Key Points

Conflict can be constructive as long as it is managed and dealt with directly and quickly. By respecting differences between people, being able to resolve conflict when it does happen, and also working to prevent it, you will be able to maintain a healthy and creative team atmosphere. The key is to remain open to other people's ideas, beliefs, and assumptions. When team members learn to see issues from the other side, it opens up new ways of thinking, which can lead to new and innovative solutions, and healthy team performance.

Things to consider:

You believe your referral would make a successful associate at Cerner. Use the Cerner Attributes as a guide. Understand that your referral is considered to be an endorsement of the candidate and your name will be tied to them as we proceed through the recruiting process.

Referral meets minimum qualifications for Cerner:

1. Authorized to work in the U.S.

2. Willingness to relocate to Kansas City

3. Ability to travel this varies greatly by role

4. Relevant experience in Business, Healthcare/Clinical, Technical/IT, Engineering

Velocity program -3.0 GPA right out of school or 2.8 GPA + 1-2 years relevant/applicable work experience

No significant gap in work history (w/o clear explanation) and no significant movement among companies in short periods of time

Please submit your referral to Referral-Recruiting@Cerner.com, with the following questions:

1. What relevant degree and/or work experience does your referral have related to Cerner?

2. Where do you think their background would be applicable at Cerner?

3. How do you know this referral/lead? If you have worked with them, share observations about their professional performance.

4. A copy of your referrals resume.

To whom it may concern,

Hi, I am Na(Natalie) Zhao and I am referring Shuowen Wei to the software engineering intern position for 2013 summer at Cerner. Shuowen is my good friend and I knew him since we took the same course Database Management System together at Wake Forest University about one year ago, when I was a 2nd year graduate student in computer science program and he was the 1st year graduate student in the mathematics program, our departments are actually in the same hall hence we met a lot.

As I know, he is getting his first master degree in mathematics this summer and he had also got admitted into the computer science program in the coming fall with a full scholarship. When I was still in Winston Salem last summer before I relocated to Kansas City and joined Cerner, I knew he was doing intern at Wake Forest Baptist Medical Center last summer as a clinic data analyst for Dr. Yaorong Ge, who was also my mentor when I was doing an intern there as software developer.

When Shuowen consulted me for summer intern opportunities in Cerner and told me he wanted to gain some working experience with developing software, I immediately suggested him to apply for the software engineer intern position, because I know he has a solid mathematics background and working experience with Oracle SQL, and he is also very good at programming, he is a smart guy and learn things fast, I believe he will be a qualified candidate for this position.

Thanks.

Best,

Natalie

Shuowen\_Wei: hi, are you Kerri?

MathWorks2: Hi

MathWorks2: Nope

MathWorks2: My name is Milind

MathWorks2: I am an Application Support Engineer in EDG

MathWorks2: How are you doing today?

Shuowen\_Wei: hi,I am a graduate student majored in CS and Maht

Shuowen\_Wei: math

Shuowen\_Wei: good, thank you, how are you today?

Shuowen\_Wei: I am looking for summer intern position at MathWorks

Shuowen\_Wei: What other positions does EDG provide other than Application Support Engineer?

MathWorks2: I am doign well.

MathWorks2: All engineers in EDG work as ASEs

MathWorks2: Have you already applied for any of the positions at MathWorks?

Shuowen\_Wei: Yes, I did

Shuowen\_Wei: may I ask what ASE does in EDG?

Shuowen\_Wei: Which programming language you use most often? C++ ?

MathWorks2: Yes

MathWorks2: Sure

MathWorks2: let me answer those questions

MathWorks2: There are two responsibilites as an ASE in EDG

MathWorks2: 1) Provide technical support to our customers as they use MATLAB and Simulink family of products.

Shuowen\_Wei: OK

MathWorks2: You work with the customers directly to debug and investigate issues they may be havinh

Shuowen\_Wei: yes, that's what I heard from Xu Xiao

MathWorks2: 2) Career Development: You work on developing your own career at MathWorks

MathWorks2: by working on internal projects

Shuowen\_Wei: then? Do you participate in developing products?

MathWorks2: The projects often include Quality Eningeering and Software Dvelopement

MathWorks2: There a lot of options based on what technology /language you like most

Shuowen\_Wei: Oh, I see

MathWorks2: MATLAB, C/C++, Java,

MathWorks2: and Web side : JavaScript, HTML 5

MathWorks2: Scripting : Perl etc.

MathWorks2: Have you already seen this page

MathWorks2: http://matlab.my/SECedg- EDG

Shuowen\_Wei: Based on my experience of using Matlab, I found Matlab is not very powerful in dealing with strings

Shuowen\_Wei: yes, I did

Shuowen\_Wei: I actually had the 2nd round techincal interview last Thursday

Shuowen\_Wei: I am currently waiting for you decision, that's why I am wondering whether all intern positons are filled so far?

Shuowen\_Wei: hello?

Shuowen\_Wei: hello?

MathWorks2: Sorry for the dealy

MathWorks2: Yes, there is no String Data type available yet.

MathWorks2: With R2013a though there are a few functions to process Strings

MathWorks2: such as STRJOIN

Shuowen\_Wei: yeah, I uesd strcat before

MathWorks2: Nope

MathWorks2: That's different

MathWorks2: Take a look at this

MathWorks2: http://www.mathworks.com/help/matlab/ref/strjoin.html

Shuowen\_Wei: I know, I never used strjoin before, my version is R2012a

MathWorks2: STRCAT simply concatanates the Char array ...it is kind of a low level functionality

MathWorks2: Anyways

Shuowen\_Wei: my version even don't have hessian

MathWorks2: So lets talk about EDG

Shuowen\_Wei: yeah,

Shuowen\_Wei: are all intern positions in EDG filled?

MathWorks2: No ..we are still hiring

Shuowen\_Wei: when do I expect any feedback or decisions from EDG? I kepte refresh my online application status everyday, it just stays the same

Shuowen\_Wei: I had techinical interview last Thursday

MathWorks2: What does it say?

Shuowen\_Wei: no further action...

Shuowen\_Wei: in the application status

Shuowen\_Wei: My friend Xu Xiao recommended me to this position

Shuowen\_Wei: I don't know whether you know him or not

MathWorks2: I don't think so

MathWorks2: Does he go by Shaw?

Shuowen\_Wei: what is Shaw? I don't know

MathWorks2: Never mind

MathWorks2: If we decide to move ahead with your application you will hear back from us within Two weeks of the date when you had the interview

Shuowen\_Wei: what I was told by that senior engineer

MathWorks2: if you do not hear back from us you can assume that we have decided not to move forward ... the online status will change indicating the same

Shuowen\_Wei: that's what I was told by that senior engineer

MathWorks2: yup

Shuowen\_Wei: I see, two weeks is just a little long, I am also having other interviews right now ...

Shuowen\_Wei: I really want to join MathWorks, one of my dream companies

Shuowen\_Wei: Anyway, may I ask you the last question?

MathWorks2: Sure

MathWorks2: You can ask as many as the time permits

MathWorks2:

Shuowen\_Wei: What if, say, I am not selected this time, since I will graduate next year, does this affect me to apply for the full time ASE position in the future? Since I plan to apply for the full time ASE position in Nov.

Shuowen\_Wei: I also plan to apply for the full time software developer position next time

Shuowen\_Wei: hello?

MathWorks2: Nope ...You can apply for other positions anytime

Shuowen\_Wei: I mean, how about this ASE position?

MathWorks2: Sorry for the delay in response ...I am talking to a lot of individuals at the same time

Shuowen\_Wei: will this affect?

Shuowen\_Wei: yeah, it's fine, I understand

MathWorks2: You can apply for the same position again after the waiting period of 6 months

Shuowen\_Wei: Oh, I see, 6 months since the day I am interviewed?

MathWorks2: Yes

MathWorks2: That's correct

MathWorks2: Do you have any other questions?

Shuowen\_Wei: nope, that's my last one, thank you Milind

MathWorks2: If you do please contact the other Engineer who is logged in as MathWorks1

Shuowen\_Wei: you have a good day!

MathWorks2: I need to go right now

Shuowen\_Wei: take care, bye

MathWorks2: Thanks for stoping by to chat

MathWorks2: Have a nice day!

Shuowen\_Wei: thanks!