What is API? Why it is important in effective programming ? Explain the implementation procedure of API in programming.

API means Application Program Interface. API is a computer interface which defines interactions between multiple software. It defines the kind of calls or requests that can be made. In other word , API is like an open language , the rules of which are shared by certain service.

You can teach your application the rules of this language, so it can communicate with the service and access all the functions and data that the service is ready to share.

For examples:

If you need to contact the news aggregator API and get ten today's most popular news from it , you refer to the 'top news' command and in response , the service will send you the latest collection. In this examples there is use of API service .

Computers make a lot of things easier, especially tasks that involve collecting and sorting through tons of data. Let's say you wanted to know how many times a particular business partner submitted invoices to your company. You could feasibly go into your company's invoice records, scan the "from" data input, and print each invoice individually for your audit.

On the other hand, if all invoices were uploaded to a central database, you could write a simple program that accesses that database and finds all the instances of the partner's name. This would take much less time and be much more accurate.

API consists of three parts:

User: the person who makes a request

Client: the computer that sends the request to the server

Server: the computer that responds to request

What are the errors that arises while implementing the algorithm?

The error that arises while implementing the algorithm are as given below:

Completeness: An algorithm is said to be complete if it definitely find solution to the problem, if exist.

Time Complexity: How long (worst or average case) does it take a find a solution? Usually measured in the terms of the number of nodes expanded.

Space Complexity: How much space is used by the algorithm? Usually measured in the terms of the maximum number of nodes in memory at a time.

Optimality: If a solution is found , is it guaranteed to be an optimal one? For examples, is it the one with minimum cost?