**LICENSING**

1. **Software**: Software, as name suggest, is simply a type of software systems that allows user to interact with computer and gives instruction to computer to perform task as well as control functioning of hardware and its operations.
2. **Algorithm**: Algorithm, as name suggests, is simply a type of process, procedure, or set of rules that must be followed to solve any type of calculations i.e., step-by-step instructions that define how work should be executed in order to get desired outcome.
3. **Differences**

|  |  |  |
| --- | --- | --- |
| **Ser** | **Software** | **Algorithm** |
|  | Software usually consists of set of data or instructions. | Algorithm usually consists of technique to solve any problem. |
|  | It is a program or any operating information that is used by computer. | It is a set of rules that must be followed in calculations. |
|  | Features of software includes efficiency, usability, portability, reliability, etc. | Features of algorithm includes input, feasibility, precision, unambiguous, easy to grasp, etc. |
|  | List of software includes google chrome, adobe reader, Microsoft office, etc. | List of algorithm includes searching algorithm, bit algorithm, recursive algorithm, randomized algorithm, etc. |
|  | Benefits of software includes increase productivity, improve data security, increase customer service, etc. | Benefits of algorithm includes easy to write, easy identification of mistakes, easy to understand logic, etc. |
|  | It takes more time to develop software. | It takes less time to create algorithm. |
|  | It is difficult to understand and create software. | It is very easy to understand and create algorithm. |
|  | Software’s are developed to meet customer requirements. | Algorithms are developed to solve any complex problems. |
|  | Design and analysis of software is important that help transformation of human readable requirements into implementation i.e., actual code. | Design and analysis of algorithm is important to solve any type of problems in computer science and information technology. |

1. **Algorithm Licensing**

Algorithms run from very simple tasks all the way up to very complex systems to analyze huge sets of data very quickly. In the case of the financial industry, it’s the ability to determine what investments to make, when and how to make them and consistently deliver a positive return on investment.

Algorithms enable the easy connection between devices in industries such as outsourced health care and medical devices, in which connected devices and sensors help diagnose and solve medical problems more effectively. Algorithms also become “smarter” as they analyze data and can determine autonomously what to do with it. One example is the diver-less car, where algorithms are learning such tasks as which way to go to avoid rear ending a car.

Because its code, algorithms are kept as trade secrets. Like other forms of intellectual property, algorithms can be licensed for a percentage of future profits. What makes algorithms so valuable to license is not the algorithm itself but the results it delivers.

1. **Types of Licenses** 
   1. **Value Based**

The practice of providing time-limited access to software or hardware features. Subscription periods can be of any length but are typically one month or one year. Software Subscriptions typically include support and maintenance, including the ability to benefit from software enhancements during the life of the Subscription. Subscription licenses may renew automatically at the end of each period or may need to be manually renewed.

* 1. **Usage Based**

A Software License Model that allows an End-User to pay for an application or device based on actual usage of that product. Consumption Based Licensing often implies the user pays a certain amount in advance and then draws down against the pre-payment based on their use (“consumption”) of the application. Consumption may be defined in terms of hours or days of use, or some other application-specific metric (such as the number of simulations run, etc.) Use of different features within the application may result in varying drawdowns against the pre-paid amount (for example, Feature A may cost X per use and Feature B may cost Y per use, etc.) Once the pre-paid amount has been exhausted the application may no longer function until the account has been topped up again (depending on vendor policy).

The practice of charging for an application based on some usage metric determined by instrumenting the application with analytics. Billing might be based on the number of hours the application is used, the number of times it is launched or some application specific metric such as the number of jobs run, format conversions processed, etc.

1. **Implementing Licenses**