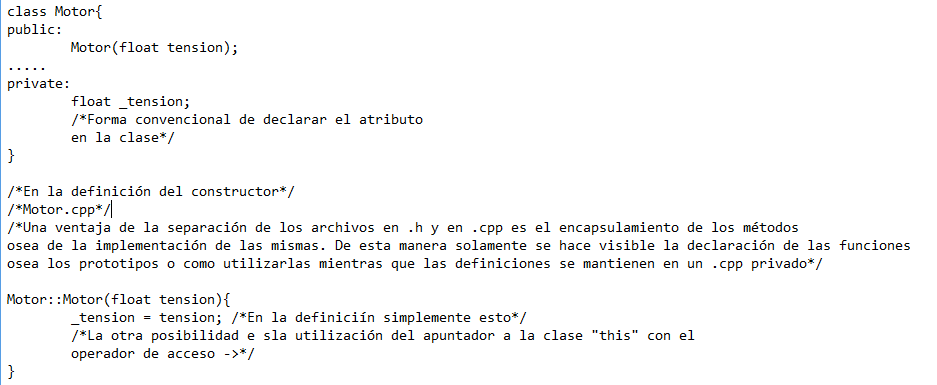
Paradigmas:

* Secuencial: ejecución en secuencia. El elemento fundamental es el dato. Un ejemplo es el código G para el cálculo numérico
* Estructurado: La unidad fundamental o la organización se centra en las estructuras y las funciones
* Orientado a objetos: El elemento principal son los objetos.

Módulo: cualquier archivo que forma parte de un proyecto o de un programa. Por ejemplo librerías, partes de código, configuración de periféricos, variables de usuario, imágenes, etc. No necesariamente tienen código ejecutable. La idea es que los módulos se compilan de forma separada lo que tiene la conveniencia de no compilar por completo la solución cuando se hace una modificación en una sola parte del mismo.

**NOTA**: Las funciones son subprogramas (no necesariamente al revés).



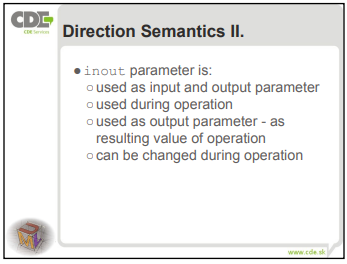
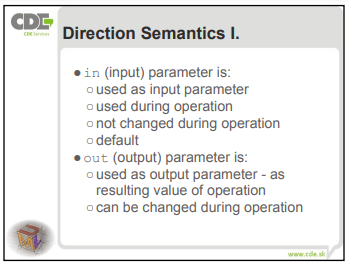
C++ tiene acceso directo a hardware. En cambio otros lenguajes de programación solamente tienen acceso a espacios virtuales de memoria.

Cuando hay una relación de dependencia, la clase u objeto dependiente no pude aportar o tener toda su funcionalidad si la existencia de la clase independiente. En una asociación en cambio el uso de la clase asociada puede ser más esporádico.

Todas las clases implementadas por el usuario en el lenguaje: C++, PYTHON, PHP, etc. heredan siempre y en la mayoría de los casos de forma implícita de una clase del lenguaje denoninada **object**. Sin embargo uno puede hacer la herencia explícita.

En Python no hay variables. O sea, las variables no son variables ni tampoco son objetos sino que son **expresiones**.

Las **funciones o métodos de hash** son para obtener índices de posición de elementos en una colección de datos. Toman como parámetro el valor de uno de los elementos de la colección y a través de un cálculo devuelven sus índices para acceder al mismo de forma directa. Se utiliza en los mapas.



El orden de las relaciones entre clases que hay que considerar en orden de prioridad es el siguiente:

* Asociaciones /dependencias
* Agregaciones/composiciones
* Herencias (no más de 3 niveles de herencia para aplicaciones de mediano a alto nivel)

**NOTA**: Las herencias presentan un problema para el recurso de la memoria.

## Anotaciones de la clase:

* Estrictamente el concepto de red de computadoras es distinto del concepto de red de datos. El primero tiene que ver con un enfoque arquitectónico mientras que el segundo es desde un enfoque de protocolos, etc.
* A los protocolos de la capa de aplicación no se accede de forma directa. En cambio las aplicaciones acceden a las funciones del protocolo, o al uso del protocolo a través del SO. Un ejemplo es la creación de un SOCKET. Una aplicación pide al sistema operativo un servicio para el acceso al protocolo y crear el socket.
* El sistema operativo (núcleo) es el que administra los recursos, la memoria y otros. Muy pocos recursos son administrados por el procesador. Por ejemplo la aplicación pide una reserva dinámica de memoria al SO
* Por lo tanto en la ejecución de una aplicación hay una permanente comunicación entre el procesador y el SO en la que el procesador va delegando tareas al SO para la administración de recursos y este va respondiendo.
* TCP/IP es un estándar de facto y Ethernet también. Pero Ethernet tomo al estándar de facto y lo normalizo. Por lo tanto hay dos protocolos, uno legal y otro de facto.
* Los protocolos de jure son legalmente establecidos y los protocolos de facto son establecidos por el uso generalizado