

Eine static Function kann ohne ein konkretes objekt aufgerufen werden.  
Sie kann nicht auf Instanzmember zugreifen.

static members sind quasi globale Variablen die wieder der Klasse zugehörig sind, nicht einzelnen objekten.

Von static class spricht man, wenn eine Klasse lediglich statische Member hat und kein Objekt von ihr erzeugt werden kann. Jedoch kann man nicht direkt ein static vor die klasse schreiben.

```
class Test
{
    public:
        static int bla;
        static void Foo();
    private:
        Test();
}
```

Zusammenfassung Text:

Die Geschichte is wichtig, da früher bereits viel konzepte erfunden wurden, die jetzt in vergessenheit geraten sind und als neu präsentiert werden.

Heutzutage sind wir nicht durch Computerleistung, sondern durch unsere Intelligenz limitiert.

Da es aber auch schlampig geht, leidet das design dadrunter.

Ende der 50er Jahre bekamen immer mehr leute die gelegenheit für compute zu programmieren. Allerdings wurde alles noch von einer riesigen maschine aus berechnet und es gab keine interaktion zwischen programmierer und computer.

Programmieren wurde schnell bekannt als eine sehr schwierige Aufgabe. Man versuchte sich mit besseren Programmiersprachen zu helfen.

1968 software crisis, systeme werden zu komplex.

Lösungsansätze: structured Programming, neue Programmiersprachen die aufgrund dessen entwickelt wurden, Pascal und ADA-

Mit Unix , dass für Minicomputers kam, wurde auch C ausgeliefert. Da die Minicomputer und Unix beliebt waren, wurde auch C immer wichtiger. Allerdings war dies eher ein Schritt zurück.

C lieferte zwar Abstraktion, was die Programme leichter verständlich machte, jedoch waren viele Abstraktionen nicht gut implementiert worden, man konnte leicht Sachen übersehen, die erst zur Runtime auftreten.

Der Massenmarkt kam ins Spiel mit den Microcomputern und als ein Texteditor, Filesystem und ein Compiler für Pascal für 50\$ angeboten wurde. Ab da konnte jeder programmieren

1972 wurde die Idee der Modularisierung geboren, eine sehr wichtige Erneuerung, machte sie doch Zusammenarbeit mehrerer Leute an einem Programm möglich.

Workstation führte Maus ein.

Objektorientierung setzte sich durch, als Windows und Personal PCs ins Spiel kamen.  
Sprachen die daraufhin entstanden: Smalltalk, Oberon, C++, Java, C#

Daraufhin folgte eine Zeit der massiven Hardware Verbesserungen und das Aufkommen des Internets hatten grossen Einfluss auf das Softwareengineering.  
OpenSource kam auf, was Unabhängigkeit anbot.

Was können wir aus der History lernen?

Wir sollten Komplexität bekämpfen wo es nur geht! Die heutigen Programmiersprachen sind noch nicht perfekt.