

## Überladen von Funktionen

- **Signatur** bezeichnet die **Anzahl**, die **Typen** und die **Reihenfolge von Parametern** einer Funktion oder Methode, der **return-Typ** gehört **nicht** zur Signatur
- ein Funktions- oder Methodenname, der von mehr als einer Funktions-/Methodendeklaration mit **unterschiedlichen Signaturen** verwendet wird, gilt als "**überladen**"
- das "**Überladen**" ist für alle Methoden einer Klasse möglich, auch für **Konstruktoren**
- die implementierten Algorithmen einer überladenen Funktion/Methode sind i.a. unterschiedlich, d.h. die Überladungen implementieren eine Funktion/Methode "**polymorph**"
- Beispiel:

```
int foo(int a, int b){ return a + b; }  
float foo(int a){ return float(a * a); }  
double foo(double a){ return 2.0 * a; }  
void main(){ cout<<foo(1, 2)<<endl;  
             cout<<foo(5)<<endl; cout<<foo(10.0)<<endl; }
```
- nicht alle Überladungen sind seitens des C++ - Compilers entscheidbar:

```
void bar(double d); void bar(long l);  
void main(){ bar(3); } //Fehler, 3 ist weder double noch long
```