



## Grundlagen der Informatik und Programmierung 2

# Neuerungen durch C++

## Überladene Funktionen

Prof. Dr. Tom Vierjahn

Visual Computing (<https://vc.w-hs.de>)  
Fachbereich Wirtschaft und Informationstechnik  
Campus Bocholt

Sommersemester 2020



main.c

```
#include <stdio.h>

int main(int argc, char** argv) {
    printf("%i\n", 42);
    printf("%f\n", 3.1415926);
    printf("%c\n", 'c');
    printf("%s\n", "GIP2");
}
```

## main.c

```
#include <stdio.h>


void print_int(int i) { printf("%i\n", i); }
void print_double(double d) { printf("%f\n", d); }
void print_char(char c) { printf("%c\n", c); }
void print_string(const char* str) { printf("%s\n", str); }

int main(int argc, char** argv) {
    print_int(42);
    print_double(3.1415926);
    print_char('c');
    print_string("GIP2");
}
```

main.c

```
void print_int(int i) { printf("%i\n", i); }  
void print_double(double d) { printf("%f\n", d); }  
void print_char(char c) { printf("%c\n", c); }  
void print_string(const char* str) { printf("%s\n", str); }
```

C-Compiler (apple clang)



Symboltabelle



## main.cpp

```
#include <cstdio>


void print(int i) { printf("%i\n", i); }
void print(double d) { printf("%f\n", d); }
void print(char c) { printf("%c\n", c); }
void print(const char* str) { printf("%s\n", str); }

int main(int argc, char** argv) {
    print(42);
    print(3.1415926);
    print('c');
    print("GIP2");
}
```

main.cpp

```
void print(int i) { printf("%i\n", i); }  
void print(double d) { printf("%f\n", d); }  
void print(char c) { printf("%c\n", c); }  
void print(const char* str) { printf("%s\n", str); }
```

C++-Compiler (apple clang++)



Symboltabelle



## Definition: Parametersignatur

Die **Parametersignatur** einer Funktion besteht aus der Anzahl, Reihenfolge, und dem Typ der übergebenen Parameter.

## Definition: Name Mangling

Der C++-Compiler benutzt **Name Mangling**, indem er den Funktionsnamen entsprechend der zugehörigen Parametersignatur eindeutig dekoriert. So kann der Linker entsprechende Funktionsaufrufe korrekt „verbinden“.

## Definition: Überladen von Funktionen

In C++ können Funktionen **überladen** werden: Eine Aufgabe (Funktionsname) kann auf zu den übergebenen Parametern passende Weise (Funktionskörper) durchgeführt werden.

# Überladene print-Funktion

das Beispiel von vorher

main.cpp

```
#include <cstdio>

void print(int i) { printf("%i\n", i); }
void print(double d) { printf("%f\n", d); }
void print(char c) { printf("%c\n", c); }
void print(const char* str) { printf("%s\n", str); }

int main(int argc, char** argv) {
    print(42);
    print(3.1415926);
    print('c');
    print("GIP2");
}
```



- ▶ C-Code
- ▶ C++-Code
- ▶ Definitionen

Prof. Dr. Tom Vierjahn

► E-Mail: [tom.vierjahn@w-hs.de](mailto:tom.vierjahn@w-hs.de)

Visual Computing

► Web: <https://vc.w-hs.de>

► YouTube: Visual Computing WH

► Twitter: [@VisComputingWH](https://twitter.com/VisComputingWH)

Westfälische Hochschule

Fachbereich Wirtschaft und Informationstechnik

Campus Bocholt



Veröffentlicht unter der Creative-Commons-Lizenz

Attribution-NonCommercial-ShareAlike 4.0 International (CC BY-NC-SA 4.0)