

# Funktionen in C++

---

Lindemann

# Verwendung von Funktionen

- Wiederkehrende Programmabschnitte werden in Funktionen ausgelagert.
- Funktionen übernehmen einen oder mehrere Werte und arbeiten mit diesen.
- Funktionen können einen Wert zurückgeben.
- Funktionen können in Funktionen aufgerufen werden. (Rekursion)

# Vorteile von Funktionen

- Steigerung der Effizienz bei der Programmierung
- Vermeidung von Fehlern durch Aufruf geprüfter Funktionen
- Verbesserung der Übersichtlichkeit des Quellcodes
- Grundlage des modularen Programmaufbaus

# Deklaration einer Funktion

Gibt an, welchen  
Datentyp der  
Rückgabewert hat.

Name der  
Funktion

Die Argumentenliste enthält  
Anzahl und Datentyp der  
Übergabewerte

Rückgabetyt Bezeichner (Parameterliste);

```
int Quadrat (int i);
```

Prototyp einer Funktion!

Hier muss ein  
Semikolon stehen!

# Definition einer Funktion

Funktionskopf

Hier darf kein  
Semikolon stehen!

Rückgabetyp Bezeichner (Parameterliste)

{

Anweisung1;

:

AnweisungN;

}

Funktionsrumpf

D  
e  
f  
i  
n  
i  
t  
i  
o  
n

```
int Quadrat (int i)
```

```
{
```

```
    i = i * i;
```

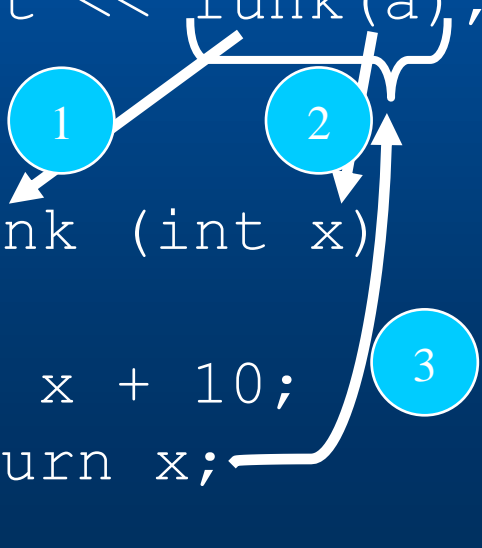
```
    return i,
```

```
}
```

Der Rückgabetyp der  
Funktion muss mit dem  
return-Datentyp  
übereinstimmen!

# Funktionsaufruf (call by value)

```
int funk (int x);  
  
int main( )  
{  
    int a = 5;  
    cout << funk(a);  
}  
  
int funk (int x)  
{  
    x = x + 10;  
    return x;  
}
```



An dieser Stelle muss funk bekannt sein (deklariert sein).

1. Aufruf der Funktion
2. Der Wert von a wird in x kopiert (5).
3. Der Funktionsaufruf wird durch den Rückgabewert ersetzt (15).