

Ein Name hat lokalen  
Geltungsbereich

A wenn er ausserhalb eines jeden  
Blockes definiert wird

B wenn er ein int-Feld bezeichnet

~~C~~ wenn er ausschliesslich  
innerhalb eines  
Schleifenblockes verwendet  
werden kann

~~D~~ wenn er innerhalb eines  
Blockes definiert wird

Ein Name hat globalen  
Geltungsbereich

~~A~~ wenn er ausserhalb eines jeden  
Blockes definiert wird

B wenn er ein char-Feld  
bezeichnet

C wenn er innerhalb eines  
Blockes definiert wird

D wenn er ausschliesslich  
innerhalb eines if-Blockes  
verwendet wird

Wozu verwendet man den Bereichsoperator :: ?

- ☒ A wenn bei gleichen Namen im globalen u. lokalen Geltungsbereich auf den globalen Namen zugegriffen wird
- ☐ B zur Rückgabe von return-Werten
- ☒ C Zugriff auf einen globalen Namen
- ☐ D Zugriff auf einen lokalen Namen
- ☐ E Zur Kennzeichnung von Konstanten

Wozu verwendet man den Bereichsoperator :: ?

- ☐ A Zugriff auf einen lokalen Namen
- ☐ B zur Rückgabe von return-Werten
- ☒ C Zugriff auf einen globalen Namen
- ☒ D wenn bei gleichen Namen im globalen u. lokalen Geltungsbereich auf den globalen Namen zugegriffen wird
- ☐ E Zur Kennzeichnung von Kommentaren

Funktionen in C++:  
welche Aussagen sind wahr

- A haben *void* als Ergebnisdatentyp, wenn kein Ergebnis zurückgegeben wird
- ~~B~~ geben immer einen Wert zurück
- ~~C~~ können andere Funktionen aufrufen
- D können max. 1 Parameter haben
- ~~E~~ können auch *int* als Ergebnisdatentyp haben.

Funktionen in C++, die kein Ergebnis zurückgeben:  
welche Aussagen sind wahr

- ~~A~~ haben *void* als Ergebnisdatentyp
- B geben immer den Wert 0 zurück
- ~~C~~ können andere Funktionen aufrufen
- D können max. 1 Parameter haben
- E können auch *int* als Ergebnisdatentyp haben.

Was bedeutet **Rekursion** in einer Programmiersprache ?

- ☒ A Eine Funktion ruft sich selbst wieder auf.
- ☐ B Das Schlüsselwort **return** muss innerhalb der Funktion vorkommen
- ☐ C Ein Aufruf der Funktion **f(x)** innerhalb der Funktion **g(x)**
- ☐ D Es muss ein Wert an die aufrufende Stelle zurückgegeben werden

Was bedeutet **Rekursion** in einer Programmiersprache ?

- ☒ A Ein Aufruf der Funktion **f(x)** innerhalb der Funktion **f(x)**
- ☐ B Es muss ein Wert an die aufrufende Stelle zurückgegeben werden
- ☒ C Eine Funktion ruft sich selbst wieder auf.
- ☐ D Das Schlüsselwort **exit** muss innerhalb der Funktion vorkommen

```
#include<iostream.h>

int f(int x){
    if (x) return f(x-4);
    else return 2;
}

int main(){
    cout << f(8);
}
```

Notieren Sie die Ausgabe

2

```
#include<iostream.h>

int g(int x){
    if (x) return g(x+3);
    else return 5;
}

int main(){
    cout << g(-6);
}
```

Notieren Sie die Ausgabe

5

Gegeben ist das Programm

```
#include<iostream.h>
```

```
int g (int a, int b){  
    cout << (a * b);  
    return a;  
}
```

```
int main(){  
    int x=2, y=4;  
    cout << g(x, y);  
}
```

Notieren Sie die Ausgabe.

82

Gegeben ist das Programm

```
#include<iostream.h>
```

```
int f (int x, int y){  
    cout << (x + y);  
    return x;  
}
```

```
int main(){  
    int a=3, b=5;  
    cout << f(a, b);  
}
```

Notieren Sie die Ausgabe.

83

```
int c(int x){  
    int etc;  
    return x;  
}
```

Welche Aussagen sind wahr?

A **x** hat einen formalen Parameter

~~B~~ **c** ist eine Funktion

C **c(4)** gibt **etc** zurück

~~D~~ **return** beendet die Funktion

~~E~~ **c** hat eine lokale Variable

~~F~~ **c(1)** gibt 1 zurück

```
int c(int k){  
    int e;  
    return 0;  
}
```

Welche Aussagen sind wahr?

A **c** ist ein formaler Parameter

B **k** ist eine Funktion

~~C~~ **k** ist ein formaler Parameter

~~D~~ **e** ist eine lokale Variable

E **c(1)** gibt den Wert von **e** zurück

~~F~~ **c(4)** gibt 0 zurück

```
#include<iostream.h>
```

```
int f(int n) {  
    return n%2;  
}
```

```
int g(int n) {  
    return n+1;  
}
```

```
int main(){  
    cout << f(g(3));  
}
```

Notieren Sie die Ausgabe.

0

```
#include<iostream.h>
```

```
int f(int n) {  
    return n+2;  
}
```

```
int g(int n) {  
    return n*2;  
}
```

```
int main(){  
    cout << f(g(4));  
}
```

Notieren Sie die Ausgabe.

10



Notieren Sie die Ausgabe des Programms.

```
#include<iostream.h>
char str[] = "JAVA";

void f(int i) {
    if (str[i])
        f(i+1);
    if (str[i])
        cout<<str[i];
}

int main() {
    f(0);
    return 0;
}
```

**A**VAJ

Notieren Sie die Ausgabe des Programms.

```
#include<iostream.h>
char str[] = "C++";

void g(int i) {
    if (str[i])
        g(i+1);
    if (str[i])
        cout<<str[i];
}

int main() {
    g(0);
    return 0;
}
```

**++**C

```
#include<iostream.h>

void f (int x){
    if (x < 2){
        cout << "jetzt";
        return;
    }
    cout << 2*x;
    return;
}

int main(){
    for(int i=2;i>0;i=i-1)
        f(i);
}
```

Notieren Sie die Ausgabe

4jetzt

```
#include<iostream.h>

void f (int x){
    if (x > 1){
        cout << "und";
        return;
    }
    cout << 2*x;
    return;
}

int main(){
    for(int i=2;i>0;i=i-1)
        f(i);
}
```

Notieren Sie die Ausgabe

und2