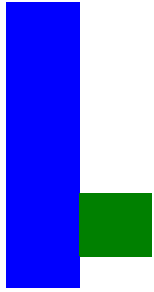


1. Grundregel

Eine Variable gilt ab der Stelle, an der man sie definiert, und bis zum Ende des Blocks aus geschweiften Klammern, in dem sie steht.

Beispiel:



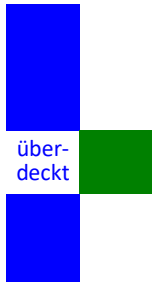
```
note neue
Note
int main()
{
    short int note;
    cin >> note;

    if (note>4)
    {
        cout << "Schade!" << endl << "Das verbessern wir:" << endl;
        short int neueNote = 2;
        cout << neueNote << endl;
    }
}
```

2. Variablen mit identischen Namen

Wenn in einem Block eine Variable vorkommt, deren Namen es auch außerhalb gibt, wird die äußere Variable vorübergehend überdeckt.

Beispiel:



```
i i
int main()
{
    int i = 3;

    if (i>1)
    {
        int i = 5;
        cout << i << endl; // Gibt 5 aus.
    }

    cout << i << endl; // Gibt 3 aus.
}
```

3. Ausnahme: Schleifen- und Funktionsköpfe

Variablen, die man im Kopf einer Schleife definiert, gelten ausschließlich in der Schleife. Sinnvoll möglich ist das - von kaum verständlichen Ausnahmen abgesehen - nur bei for-Schleifen.

Beispiel:

```

i      i
      int main()
      {
          for (int i=1; i<=5; i++)
          {
              cout << i << endl;
          }

          for (int i=17; i<=21; i++)
          {
              cout << i << endl;
          }
      }

```

Ausblick auf das nächste Thema: Parameter, die in der Kopfzeile einer Funktion stehen, gelten innerhalb der Funktion als Variablen.

Empfehlung

Variablen, die man nur kurz benötigt, definiert man am besten an Ort und Stelle statt am Programmstart. Dadurch ist der Code leichter durchschaubar und das Programm benötigt vielleicht sogar weniger Speicher.

Aufgaben

1. Erklären Sie, welchen Fehler das folgende Programm enthält!

```

int main()
{
    for (int i=0; i<10; i++)
    {
        cout << i << endl;
    }
    cout << i << endl;
}

```

2. Geben Sie an, welche Bildschirmausgabe der folgende Programmausschnitt erzeugt!

```

int a=7;
{
    int a=0;
    for (int a=1; a<=3; a++)
        cout << a << " ";
    cout << a << " ";
}
cout << a << " " << endl;

```
