

	Beispiel	Erläuterung des Beispiels	Kommentar
const	const int x = 5;	x ist eine Konstante. Ihr Wert kann nicht verändert werden.	
Referenz-Variable	int & x ;		Nicht behandelt
Kommentar	int anz ; // Anzahl	Rest der Zeile nach // gilt als Kommentar	
Parameter-übergabe	f (const int x , int & y){ x = 3 ; //nicht zulässig y = x; //zulässig }	<ul style="list-style-type: none"> • x wird per Wert übergeben. Der übergebene Wert kann nicht verändert werden. • y wird per Referenz (Adresse) übergeben. Der Wert von y wird mit der Anweisung y = 3; verändert. 	f (const int r , int * s); r = 7 ; //zulässig. // Ändert den // übergebenen // Wert *s = 5 ; //zulässig. Ändert //den Wert von s
Standard-Ein-Ausgabe	int x = 4 ; cout << x ; cin >> x ;	<ul style="list-style-type: none"> • Der Wert von x wird auf Bildschirm geschrieben. Dabei wird ein Standard-Format genutzt. • Ein neuer Wert für x wird von Tastatur eingelesen. 	Vergleichbar mit: <ul style="list-style-type: none"> • printf • scanf
break		Benutzung nur in switch-Anweisungen zugelassen, sonst verboten.	

- Keine implizite Deklaration von Funktionen, Typen, Parametern usw. Alles muss explizit deklariert werden.
- C++ lässt lokale Variable in Schleifen zu, z.B.

```
for (int i;i<4;i++)
{....}
```
- Definition von Variablen überall im Programm zugelassen (nicht nur vorne vor Anweisungen).
- Mit typedef werden Datentypen definiert. struct dient auch zur Typdefinition.
- struct wird nur bei Typdefinition benutzt. Nicht aber an anderen Stellen, wie bei der Datenobjektdefinition.
- Man kann mit enum Aufzählungstyp definieren.
- Es gibt implizit den Datentyp bool mit den Werten false und true.
- Beim Aufruf einer Funktion müssen die Anzahl und Typen der aktuellen Parametern mit denen der formalen Parameter übereinstimmen.
- Zwei Funktionen mit demselben Namen dürfen vorkommen, wenn sie sich in Anzahl bzw. Typ ihrer Parameter unterscheiden.
- Mixed Mode Arithmetik nicht zulässig. Explizit casten.