Datentypen

`unsigned int`	Datentyp für **natürliche Zahlen** (inklusive 0)
Literal: `...u`	

```
unsigned int a = 4;   // Conversion int --> unsigned int
unsigned int b = 4u;  // No conversion
std::cout << a - 5 << "\n";  // too small (underflow)
```

Operatoren

`/`	**Division**
Präzedenz: 14 und Assoziativität: `links`	
Falls `int`s oder `unsigned int`s dividiert werden, so rundet der Operator automatisch zu 0 hin.	

```
unsigned int a = 9 / 3;  // Result:  3
unsigned int b = 5 / 3;  // Result:  1
int c = -3 / 2;          // Result: -1
```

`%`	**Modulo**. Rest der *Ganzzahl*division
Präzedenz: 14 und Assoziativität: `links`	
`%` gibt es *nur* für `int` und `unsigned int`. Bei negativen Zahlen übernimmt `%` das Vorzeichen des linken Operanden.	

```
int a = 5;
int division = a / 3;    // Result:  1
int rest = a % 3;        // Result:  2
int negative = -5 % -3;  // Result: -2
```

`++...`	**Prä-Inkrement**. Erhöht den Wert der Variablen und gibt den *neuen* Wert zurück.
Präzedenz: 16 und Assoziativität: `rechts` Sonst gibt es noch: `--...` **Prä-Dekrement**	

`int a = 0;`
`int b = ++a; // b gets value 1,`
`// a gets value 1`	

`...++`	**Post-Inkrement**. Erhöht den Wert der Variablen und gibt den *alten* Wert zurück.
Präzedenz: 17 und Assoziativität: `links` Sonst gibt es noch: `...--` **Post-Dekrement**	

`int a = 0;`
`int b = a++; // b gets value 0,`
`// a gets value 1`	

`+=`	**Addiert** den **rechten** Operanden **zum linken** Operanden.
Präzedenz: 4 und Assoziativität: `rechts` Sonst gibt es noch: `-=...` für **Subtraktion** `*=...` für **Multiplikation** `/=...` für **Division** `%=...` für **Modulo**	
`int a = 4;` `a += 5;  // a gets value 9`	

Generell

`std::numeric_limits<T>::min()`	Ermittelt **kleinsten zulässigen Wert** des Datentyps `T`.
Erfordert: `#include <limits>` Sonst gibt es noch: `std::numeric_limits<T>::max()`	

```
int lower_bdd = std::numeric_limits<int>::min();
std::cout << "Enter a number larger than " << lower_bdd << ": ";
int input;
std::cin >> input; // User knows the smallest valid number.
```