

## Veranstaltungsseite:

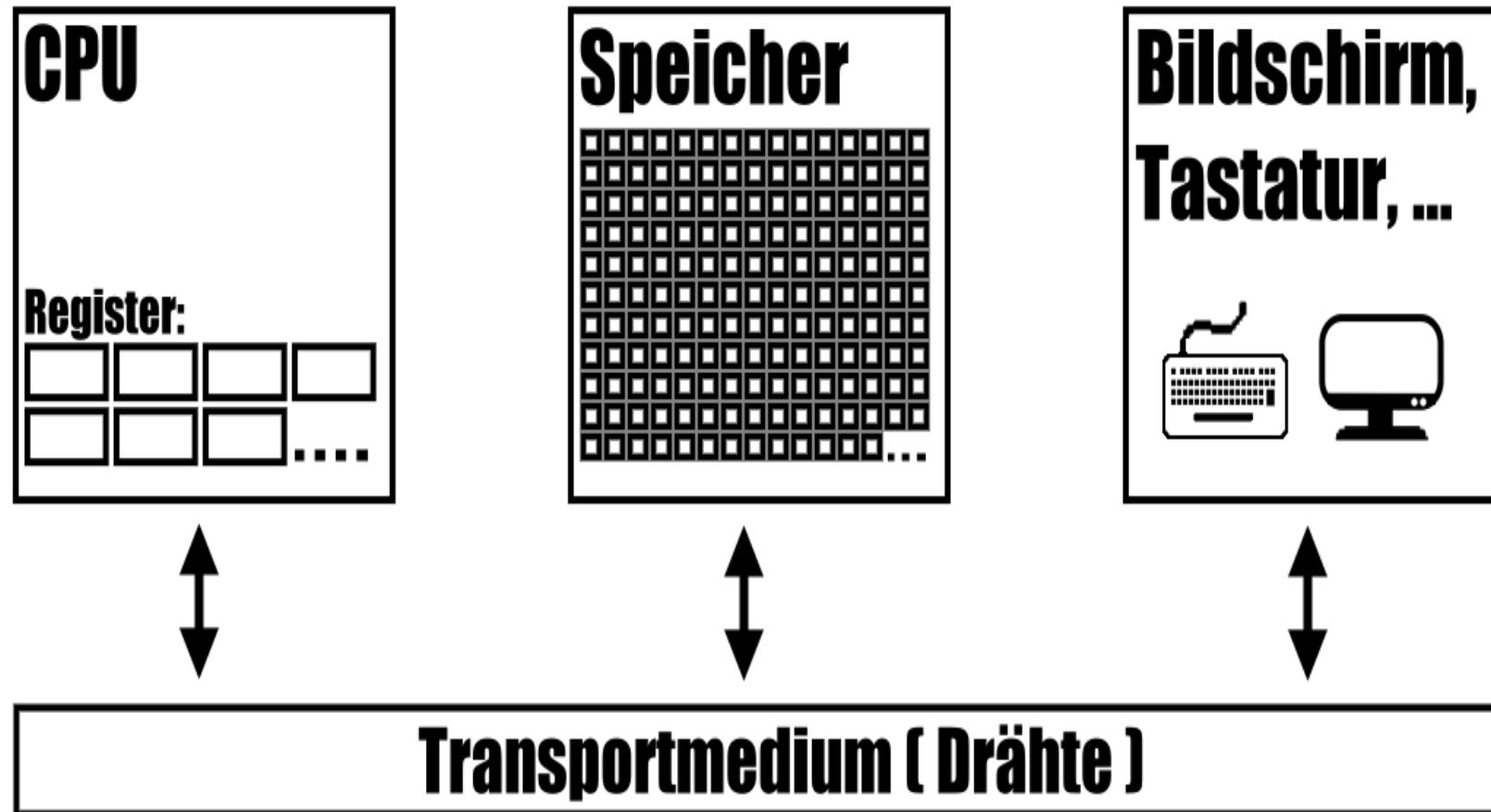
- [https://precampus.uni-bonn.de/login.php?target=crs\\_78&cmd=force\\_login&lang=de](https://precampus.uni-bonn.de/login.php?target=crs_78&cmd=force_login&lang=de)
- Auch hier auffindbar hier:  
[www.or.uni-bonn.de/lectures/ws24/programmierkurs\\_ws24.html](http://www.or.uni-bonn.de/lectures/ws24/programmierkurs_ws24.html)
- **Zeitraum:** 23. September 2024 bis 4. Oktober 2024 (nicht am 3. Oktober)
- **Vorlesungen:** 10 Uhr c.t. (also 10:15 Uhr) bis 11:45 Uhr
- **Übungen:** 13 Uhr c.t. bis 17 Uhr
  - im Großen oder Kleinen Hörsaal,
  - im PC-Pool im Mathematikzentrum, Endenicher Allee 60, Nebengebäude, Raum N0.004 / N0.005
  - im PC-Pool Wegelerstraße 6 (Hochschulrechenzentrum)
- Es gibt jeden Tag einen Übungszettel, der in den Übungen bearbeitet werden soll.

`https://precampus.uni-bonn.de/login.php?target=crs_78&cmd=force_login&lang=de`

## Vorlesungsunterlagen

- Auf der Precampus-Seite (s.o.) finden Sie
  - ein fortlaufend aktualisiertes Skript,
  - die Folien aus der Vorlesung,
  - die Beispiel-Programme aus der Vorlesung
  - die Übungszettel
- Eine Beschreibung, wie man eine **Programmierungsumgebung** unter Windows und Mac OS X einrichtet, steht im Skript.

# Schematischer Aufbau des Computers



- Die Berechnung eines Computers laufen in der **CPU** (Central Processing Unit)
- Die CPU enthält eine kleine Anzahl von Speicherzellen (**Registern**), die z.B. je 64 Bits speichern können.
- Berechnungen müssen in **elementare Rechenoperationen** zerlegt werden, z.B.
  - Addition der Inhalte von zwei Registern und Speicherung des Ergebnisses in einem dritten Register
  - Kopieren eines Registerinhalts an eine bestimmte Stelle im Hauptspeicher.
- Solche Befehl heißen **Maschinencode**.
- ⇒ Ungeeignet für die Entwicklung komplexer Programme

Ein **Compiler** übersetzt für Menschen lesbaren Code (z.B. ein C++-Programm) in für Computer lesbaren Maschinencode.

## Aufruf:

```
g++ -std=c++11 -Wall -Wpedantic -o EXECUTABLE QUELLDATEI
```

## Erklärung der Komponenten:

- `g++`: Name des Compilers
- `-std=c++11`: Genaue Version von C++ (in diesem Fall Version 11)
- `-Wall`: Alle Warnungen werden berücksichtigt.
- `-Wpedantic`: Der Compiler soll kleinlich sein.
- `-o EXECUTABLE`: Name des erzeugten Maschinencode-Programms
- `QUELLDATEI`: Name der Datei mit dem C++-Code

# Die Standardbibliothek

- Stellt viele bereits programmierte Funktionalitäten zur Verfügung.
- Namen von Dinge aus der Standardbibliothek beginnen mit `std::`, also z.B. `std::cout` oder `std::endl`
- Schnittstellen zu den Funktionen werden in Header-Dateien wie z.B. `iostream.h` definiert. Die Einbindung erfolgt z.B. mit `#include <iostream>`
- Die Module der Standardbibliothek sind bereits vorab kompiliert.

# Variablen

Eine Variable kodiert zwei Dinge:

- einen **Speicherabschnitt**
- einen **Datentyp**, der sagt, wie die 0-1-Folge in diesem Abschnitt zu interpretieren ist (als Zahl, Buchstabe etc.)

Syntax:

```
DATENTYP NAME = WERT;
```

Wichtige Datentypen:

- `int`: speichert eine ganz Zahl
- `double`: speichert auch Brüche

Beispiele:

- `int x = 37;`
- `double y = 2.8;`

# Konstanten

## Syntax:

```
const DATENTYP NAME = WERT;
```

Der Wert der Variablen kann danach nicht mehr geändert werden.

Das folgende Beispiel ist daher kein korrekter C++-Code:

```
const double pi = 3.14159;  
pi = 2.71; //Das geht nicht!
```



Umwandlung zwischen Datentypen folgt bei Berechnungen oft automatisch.

Syntax für manuelle Umwandlung:

```
static_cast<DATENTYP> (AUSDRUCK) ;
```

**Beispiel:**

```
double x = 12.7;  
int y;  
y = static_cast<int>(x);
```

# Kurzschreibweisen

Für eine Variable `x` (z.B. ein `int`) ist `x += a;` eine Kurzschreibweise für `x = x + a;`.

## Beispiel:

```
int x = 37;  
int y = 5;  
x += y;
```

Anschließend hat `x` den Wert 42.

## Analog:

- `x -= 7;`
- `x *= y;`
- `x /= 2;`

# Unäre Operatoren

Um den Wert einer Variable  $x$  um 1 zu erhöhen kann man

$x += 1;$

schreiben.

Mit dem Ausdruck  $++x$  erreicht man das auch.

Analog:

$--x$  statt  $x -= 1$  bzw.  $x = x - 1$