
Stoomcursus C++



Universiteit
Leiden

Walter Kusters

februari 2023

[video](#)

www.liacs.leidenuniv.nl/~kusterswa/AI/

Als je thuis gratis C++ wilt doen, zijn de mogelijkheden:

- Windows: **Code::Blocks**, zie straks

- Mac: zie [hier](#)

[Mac-video](#)

- Linux: haal **Ubuntu** van www.ubuntu.org
huidige versie: 22.10 (in computerzalen: 18.04)
gebruik editor **gedit** en compiler **g++**

[Linux-video](#)

- Windows10/11: installeer Windows Subsystem for Linux (WSL) en editor [Sublime Text](#)

We gebruiken in het begin vooral **Code::Blocks** in Windows, daarnaast ook g++ en gedit in Linux.

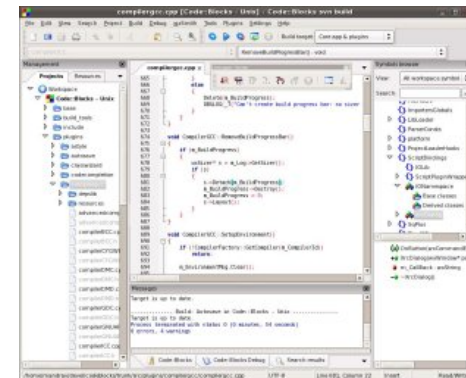
[video](#)

- installeer Code::Blocks van www.codeblocks.org
file: codeblocks-20.03mingw-setup.exe

- zet “warnings” aan!

- edit, compileer en run
helloworld.cc (let op .cc)

- behandel een eventuele lastige firewall



Een werkend C++-programma maken gaat als volgt:

1. Tik in een **editor** C++-code, het “bron-programma”.
2. Compileer (en link) dit met een **compiler**. Deze vertaalt C++-code naar machinetaal.
Eventuele fouten: “compile-time-fouten” / “syntax-errors”: “inklude” in plaats van “include”.
3. Draai (= run) deze **executable** vanuit het OS.
Eventuele fouten: “run-time-fouten”: deel door 0.

Herhaal deze cyclus zo vaak als nodig.

In Code::Blocks vinden alle stappen plaats in een Graphical User Interface (GUI).

(Er zijn ook nog logische fouten.)

Een eerste C++-programma, helloworld.cc geheten:

```
#include <iostream>
using namespace std;
int main ( ) {
    cout << "Dag allemaal!" << endl;
    return 0;
} //main
```

Dit programma zet alleen een tekstje op het beeldscherm.

Let op de — vooral voor mensen nuttige — **layout**.

En op hoofdletters en kleine letters.

Een tweede C++-programma (met **syntax-highlighting**):

```
1 // dit is een simpel programma
2 #include <iostream>
3 using namespace std;
4
5 int main ( ) {
6     int getal = 42; // een variabele declareren
7                     // en initialiseren
8     cout << "Geef een geheel getal .. ";
9     cin >> getal;
10    cout << "Kwadraat is: "
11         << getal * getal << endl;
12    return 0;
13 } // main
```

```
// Dit is een regel met commentaar ...
#include <iostream> // moet er altijd bij
using namespace std;
const double pie = 3.14159; // een constante (of cmath)
int main ( ) {
    int straal; // straal van de cirkel, geheel getal
    cout << "Geef straal, daarna Enter .. ";
    cin >> straal;
    if ( straal > 0 )
        cout << "Oppervlakte "
            << pie * straal * straal << endl;

    else
        cout << "Niet zo negatief ..." << endl;
        cout << "Einde van dit programma." << endl;
    return 0;
} //main
```

```
// Dit is een regel met commentaar ...
#include <iostream> // moet er altijd bij
using namespace std;
const double pie = 3.14159; // een constante (of cmath)
int main ( ) {
    int straal; // straal van de cirkel, geheel getal
    cout << "Geef straal, daarna Enter .. ";
    cin >> straal;
    if ( straal > 0 ) { // accolades nodig!
        cout << "Oppervlakte ";
        cout << pie * straal * straal << endl;
    }//if
    else
        cout << "Niet zo negatief ..." << endl;
    cout << "Einde van dit programma." << endl;
    return 0;
} //main
```


In een [Linux](#) windows-omgeving zoals KDE of GNOME start je een of meer **terminals**: windows waarin je tekst-georiënteerde opdrachten kunt geven. Daarin tik je in:

<pre>gedit een.cc &</pre>	edit je eerste C++-programma; open hiertoe een "edit-window"
<pre>g++ -Wall -o een een.cc</pre>	compileer een.cc naar een
<pre>./een</pre>	run de executable een
<pre>ls -lrt</pre>	overzicht van je files (%)

De ampersand & zorgt er voor dat je in het oorspronkelijke window ook kunt doorwerken. En (%) levert zoiets als:

```
-rw-r--r-- 1 kosterswa domain users 124 Sep 6 12:52 een.cc
-rwx----- 1 kosterswa domain users 11049 Sep 6 12:53 een
```

In C++ heeft elke variabele een vast type:

```
int i;           // integer = geheel getal
bool b;         // Boolean: true OF false
char c;         // karakter, bijvoorbeeld 'm' of '7' of '$'
double gewicht; // "reeel" getal
int A[10];      // array met 10 integers: A[0],A[1],...,A[9]
```

Een if-statement:

```
if ( x > 0 ) {
    ...
} //if
else {
    ...
} //else
```

Let op haakjes, accolades en inspringen.

Standaard for-loop en while-loop:

```
for ( i = 0; i < n; i++ ) {  
    ...  
} //for
```

```
while ( x > 0 ) {  
    ...  
} //while
```

Let wederom op haakjes, accolades en inspringen.

Er zijn twee soorten functies:

```
void telop (int x, int y, int & z) {  
    z = x + y;  
} //telop
```

```
int kwadraat (int w) {  
    return w * w;  
} //kwadraat
```

De parameter-overdracht is doorgaans **call-by-value**. Als er een **&** (“ampersand”) bij staat is het **call-by-reference**. Na aanroep `telop (3,4,k)`; heeft `k` de waarde 7. En met `x = kwadraat (k)`; wordt `x` nu 49.

Zie verder de voorbeeldcode in:

www.liacs.leidenuniv.nl/~kosterswa/AI/clobber2023.cc

Bekijk eerst de klasse `clobber` en een paar memberfuncties, zoals `aantalzetten ()` en `randomspel ()`.

Veel meer informatie, video's, . . . , is te vinden bij de website van het vak Programmeermethoden:

www.liacs.leidenuniv.nl/~kosterswa/pm/