

# 1 Hello, World!

Volgens oude traditie is ons eerste C++-programma een “Hello, World!”-applicatie. We gaan deze applicatie op de commandolijn compileren en linken.

Om dat te kunnen, moet je eerst een toolchain op je computer installeren. Welke toolchains er beschikbaar zijn, hangt af van je besturingssysteem. Hieronder wordt er per besturingssysteem een (incomplete) lijst van opties gegeven.

## 1.0.0.1 Windows

Voor Windows kun je Visual Studio 2015 installeren, let erop dat deze standaard geen C++-compiler installeert. Je zult tijdens de installatie moeten aangeven dat je in Visual C++ wilt programmeren. Als alternatief kun je de Microsoft Visual C++ Build Tools installeren. In het Startmenu vind je nu volgende items:

- **32-bits:** VS2015 x86 Native Tools Command Prompt
- **64-bits:** VS2015 x64 Native Tools Command Prompt

Als alternatief kun je Cygwin installeren. Cygwin is een soort linux-distributie voor Windows. Deze levert de g++-compiler.

## 1.0.0.2 macOS

Installeer Xcode uit de App Store. Start Xcode en accepteer de licentie, Xcode zal nu een aantal componenten op je systeem installeren. Voer als laatste stap, volgend commando in een terminalvenster:

```
xcode-select --install
```

## 1.0.0.3 Linux

Installeer gcc of clang. Hoe dit moet, verschilt per distributie. Hieronder vind je instructies voor Debian en Fedora.

### 1.0.0.3.1 Debian

Om de gcc-toolchain te installeren:

```
sudo apt-get install build-essential
```

Om de clang-toolchain te installeren onder Jessie

```
sudo apt-get install clang-3.5
```

### 1.0.0.3.2 Fedora

Om de **gcc**-toolchain te installeren:

```
sudo dnf group install "C Development Tools and Libraries"
```

Om de **clang**-toolchain te installeren:

```
sudo dnf install clang
```

## 1.1 Give me the code!

Lees nu pagina's 37 t/m 39 in "The C++ Programming Language".

Schrijf nu het Hello, World! programma op bladzijde 39 in een tekstbestand `hello.cpp`. We gaan dit bestand compileren en linken vanaf de commandolijn.

## 1.2 Preprocessen

We gaan nu het resultaat van de preprocess stap bekijken. Gebruik de instructies van jouw toolchain om `hello.cpp` te preprocessen tot `hello.i`. Maak vervolgens deze opdracht:

1. Wat is er gebeurt met het bestand? Verklaar wat de preprocessor heeft gedaan.

### 1.2.0.1 VS2V015 Build Tools

Ga in de command prompt naar de map met `hello.cpp`, voer daar volgend commando uit:

```
cl /P hello.cpp
```

`cl` plaats het resultaat in het bestand `hello.i`.

### 1.2.0.2 Clang (Linux / macOS)

Ga in je terminal naar de map met `hello.cpp`, voer daar volgend commando uit:

```
clang++ -E hello.cpp -o hello.i
```

### 1.2.0.3 GCC (Linux / Cygwin)

Ga in je terminal naar de map met `hello.cpp`, voer daar volgend commando uit:

```
g++ -E hello.cpp -o hello.i
```

## 1.3 Compileren

We gaan `hello.cpp` nu compileren tot een object file

### 1.3.0.1 VS2015 Build Tools

```
cl /c /EHsc hello.cpp
```

`cl` plaats het resultaat in het bestand `hello.obj`.

### 1.3.0.2 Clang (Linux / macOS)

Ga in je terminal naar de map met `hello.cpp`, voer daar volgend commando uit:

```
clang++ -c hello.cpp
```

### 1.3.0.3 GCC (Linux / Cygwin)

Ga in je terminal naar de map met `hello.cpp`, voer daar volgend commando uit:

```
g++ -c hello.cpp
```

## 1.4 Linken

Als laatste stap in het process moeten we het bestand linken, doe daartoe het volgende:

### 1.4.0.1 VS2015 Build Tools

```
cl hello.obj
```

`cl` plaats het programma in het bestand `hello.exe`.

### 1.4.0.2 Clang (Linux / macOS)

Ga in je terminal naar de map met `hello.cpp`, voer daar volgend commando uit:

```
clang++ -o hello hello.o
```

`clang++` maakt nu het uitvoerbare bestand `hello` aan. Voer het uit met volgende instructie:

```
./hello
```

### 1.4.0.3 GCC (Linux / Cygwin)

Ga in je terminal naar de map met `hello.cpp`, voer daar volgend commando uit:

```
g++ -o hello hello.o
```

`g++` maakt nu het uitvoerbare bestand `hello` aan. Voer het uit met volgende instructie:

```
./hello
```

## 2 IDE's, to each his own.

Installeer en compileer het “Hello, World!”-programma in twee verschillende C++ IDE's. Hieronder staan een aantal voorbeelden.

- CLion (Cross-platform)
- Code::Blocks (Cross-platform)
- GNOME Builder (Alleen op recente Linuxdistributies, geen geïntegreerde debugger)
- Eclipse CDT (Cross-platform)
- Qt Creator (Cross-platform, gericht op Qt-applicaties)
- Visual Studio (Alleen Windows)