

# Hello World!

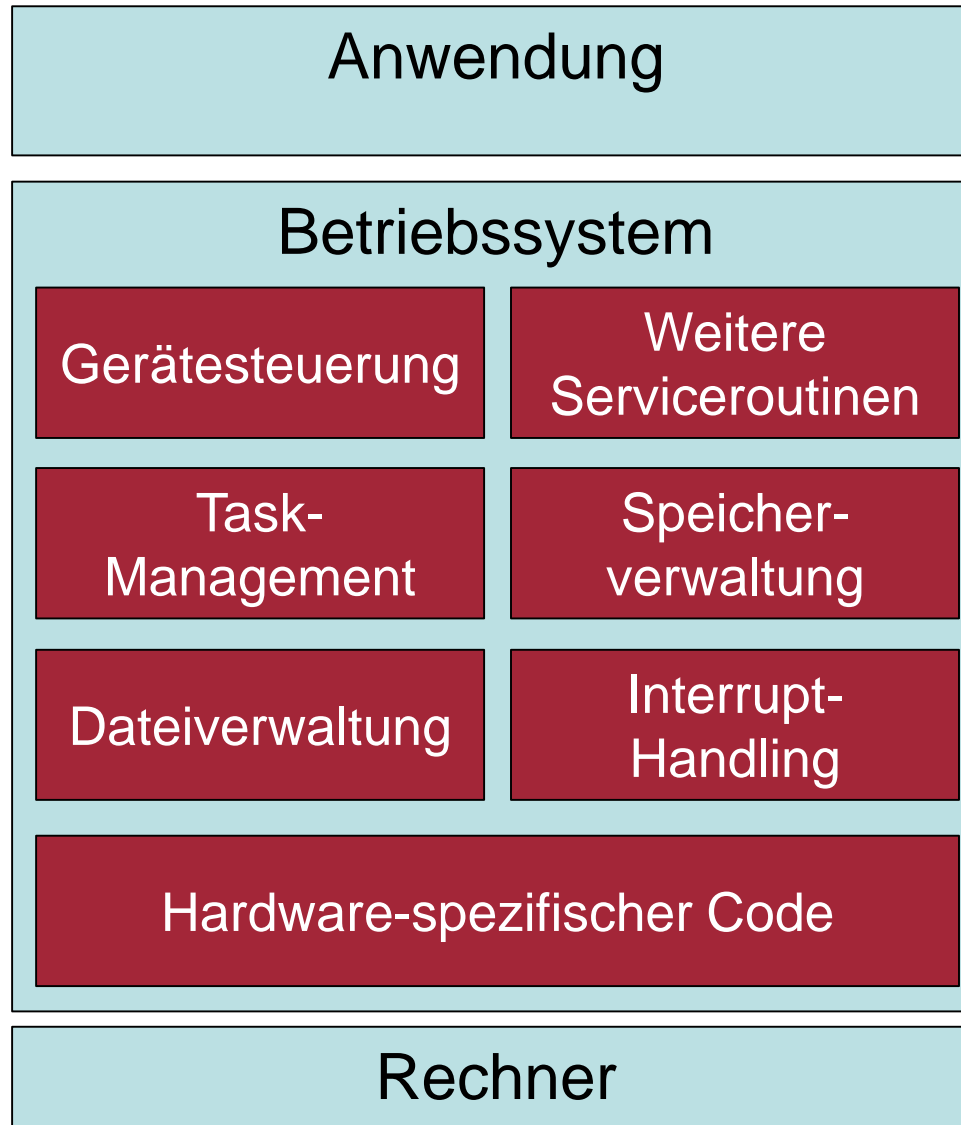
## Eine Einführung in das Programmieren

# Das erste Programm

Görschwin Fey

Institute of Embedded Systems  
Hamburg University of Technology

# Betriebssystem – Funktion



# Programmierung

## Rechner

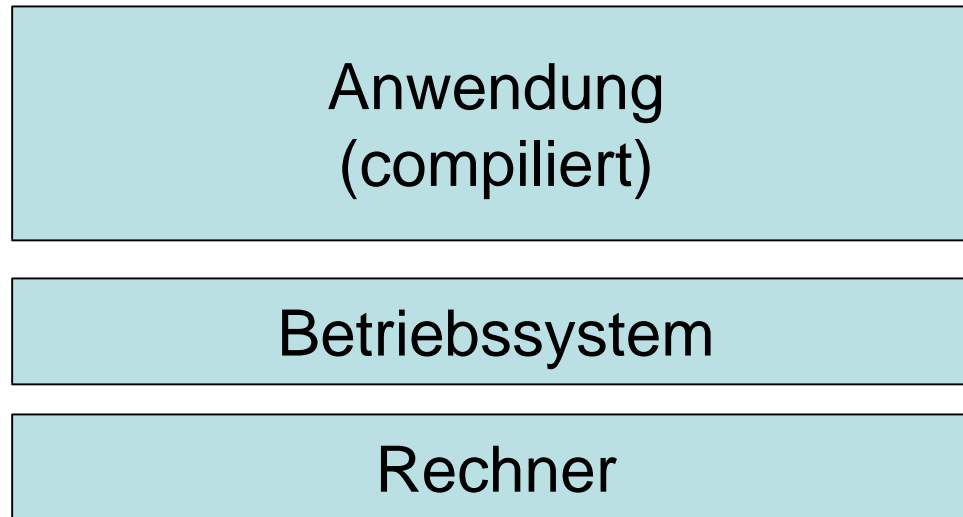
- Stellt nur sehr einfache Befehle zur Verfügung
- Programm
  - ist im Speicher abgelegt, muss in Binärformat vorliegen, sogenannter *Maschinencode*
  - muss für das jeweilige Betriebssysteme angepasst sein

## Programmierung

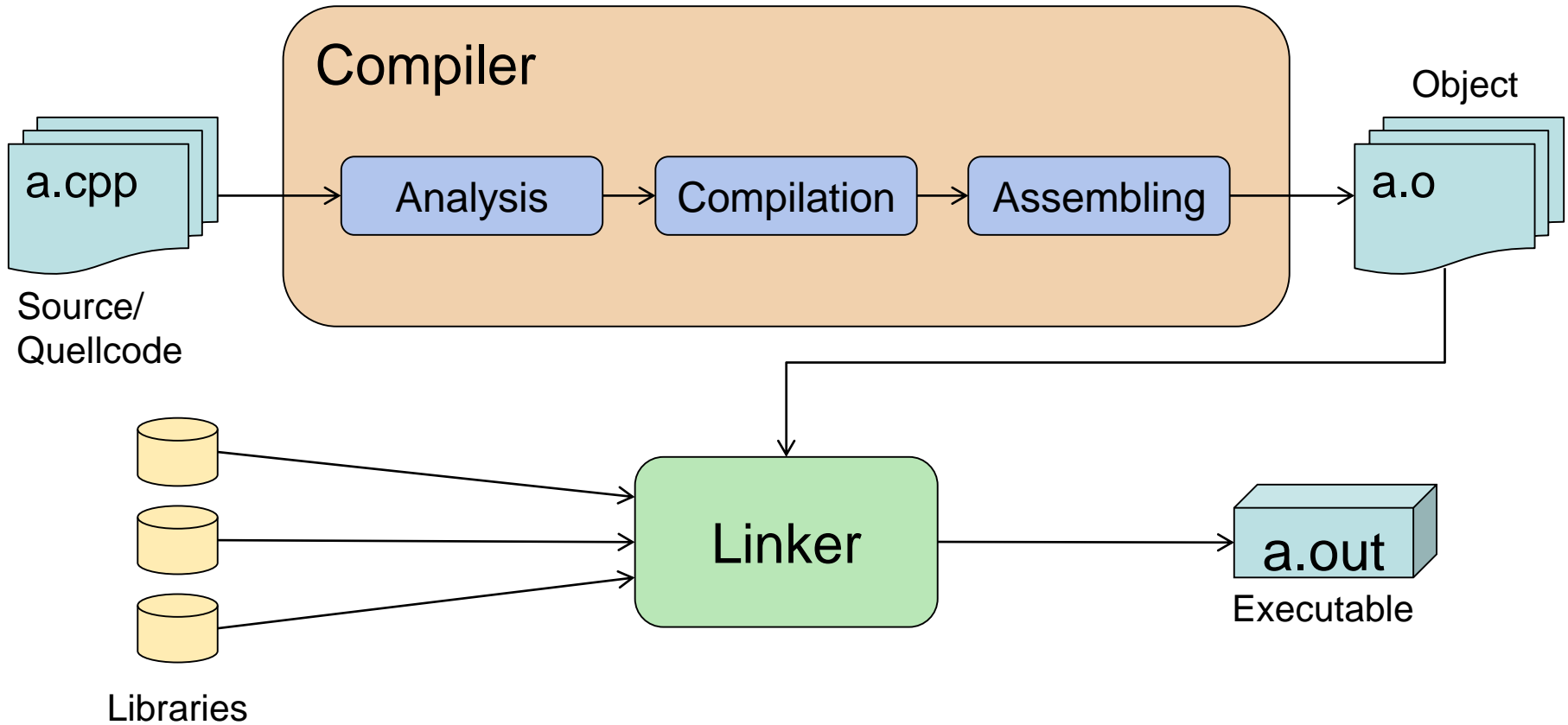
- Geschieht (heute fast immer) in einer sogenannten Hochsprache
  - Programm ist eine Textdatei
- Übersetzung in *Assembler* und von dort in Maschinencode passiert mit Hilfssoftware:
  - Compilierung

# Übersetzung von Hochsprache in Maschinencode

- Compilieren: Programm wird vor der Ausführung übersetzt



# Compiler-Vorgang



- Vereinfachte Darstellung des Compiler-Vorgangs
- Bibliotheken (Libraries) implementieren zusätzliche Funktionalität
- Linker verbindet das Programm mit den Bibliotheken zum Executable

# Werkzeuge

- `g++`
  - C++-Compiler & -Linker
  - Unter Linux oft vorinstalliert bzw. leicht verfügbar
  - Unter Windows als `mingw` bzw. in Codeblocks verfügbar
  - Unter MacOS mit `xcode` verfügbar
  - Kommando im Terminal

```
g++ <SourceCode>
```

- `<SourceCode>` ist eine Textdatei mit C++-Anweisungen
- erzeugt das Programm `a.out`
- Programm wird durch Terminaleingabe ausgeführt

```
./a.out
```

# Anweisungen in C++

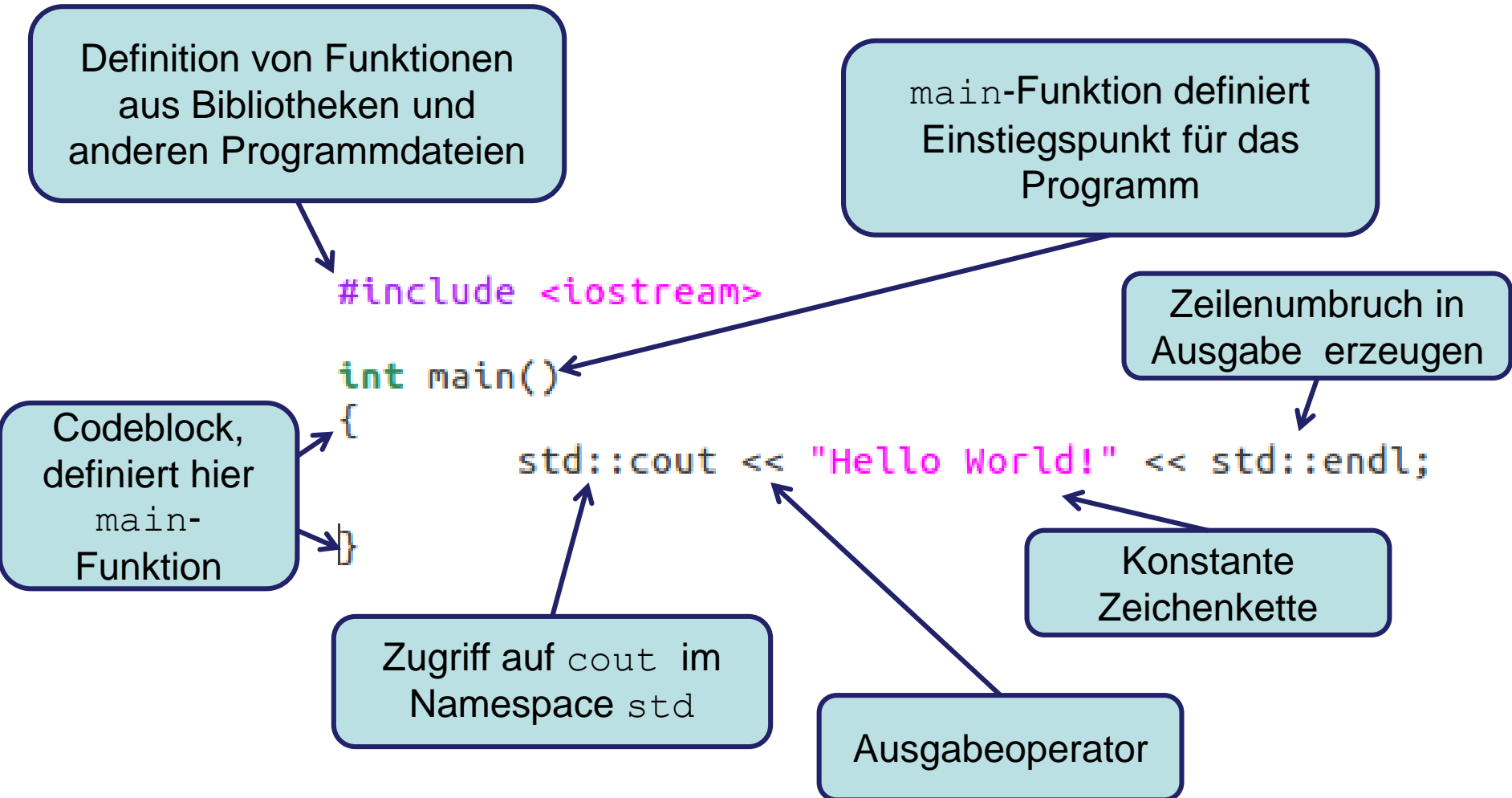
Jede Programmanweisung ...

- endet mit einem Semikolon
- wird in eine neue Zeile geschrieben

```
Anweisung 1;  
Anweisung 2;  
...  
Anweisung n;
```

Anweisungen eines Programms werden sequentiell nacheinander ausgeführt

# Hello World – Ein erstes C++-Programm: Datei a.cpp





# Syntax und Semantik

- Syntax – *wie die Grammatik einer Sprache*
  - Regeln, die beschreiben, welche Zeichenfolgen gültige Programme sind
- Semantik – *wie die Bedeutung der Worte einer Sprache*
  - Bedeutung, welche Aktionen durch einzelne Befehle ausgeführt werden
- Compiler
  - kennt die Syntax
  - prüft, ob die Regeln eingehalten sind
    - falls ja, wird ein Programm erzeugt dessen Semantik wohldefiniert ist
    - falls nein, liefert der Compiler Fehlermeldungen als Hinweise

# Zusammenfassung

- Betriebssystem
  - Vereinfacht die Nutzung der Hardware
  - Erlaubt gewisse Abstraktion von Hardware
- Hochsprachen werden mit einem Compiler jeweils für Betriebssystem und Hardware übersetzt
- Wir benutzen g++ als Compiler für die Hochsprache C++

# Übung: Hello World programmieren

1. Textdatei hello.cpp erzeugen mit diesem Inhalt (Verzeichnis merken!)

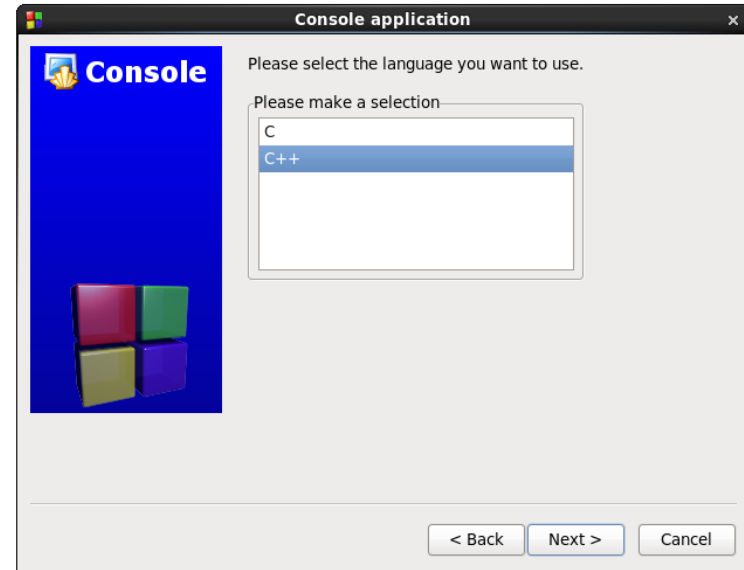
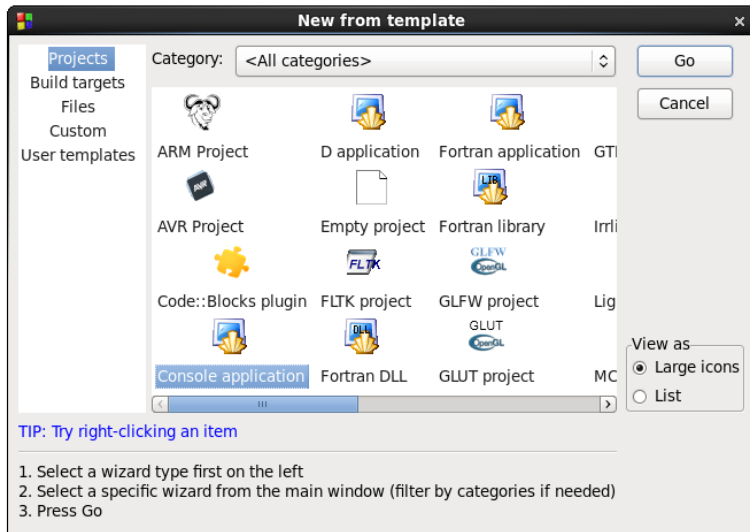
```
#include <iostream>
int main()
{
    std::cout << "Hello, World!\n";
    return 0;
}
```

2. Programm mit Compiler g++ per Anweisung im Terminal compilieren:  
g++ hello.cpp
3. Programm wurde als Datei a.out bzw. a.exe erzeugt (ls)
4. Programm ausführen mit Kommando ./a.out
5. Fehler in das Programm einbauen, erneut compilieren, schauen, was der Compiler ausgibt, ggf. Programm ausführen  
Z.B. „;“ am Zeilenende entfernen; „<“ entfernen, „\n“ entfernen, Text „Hello World“ ändern
6. Programm korrigieren. Den Text Hello, World! zweimal ausgeben

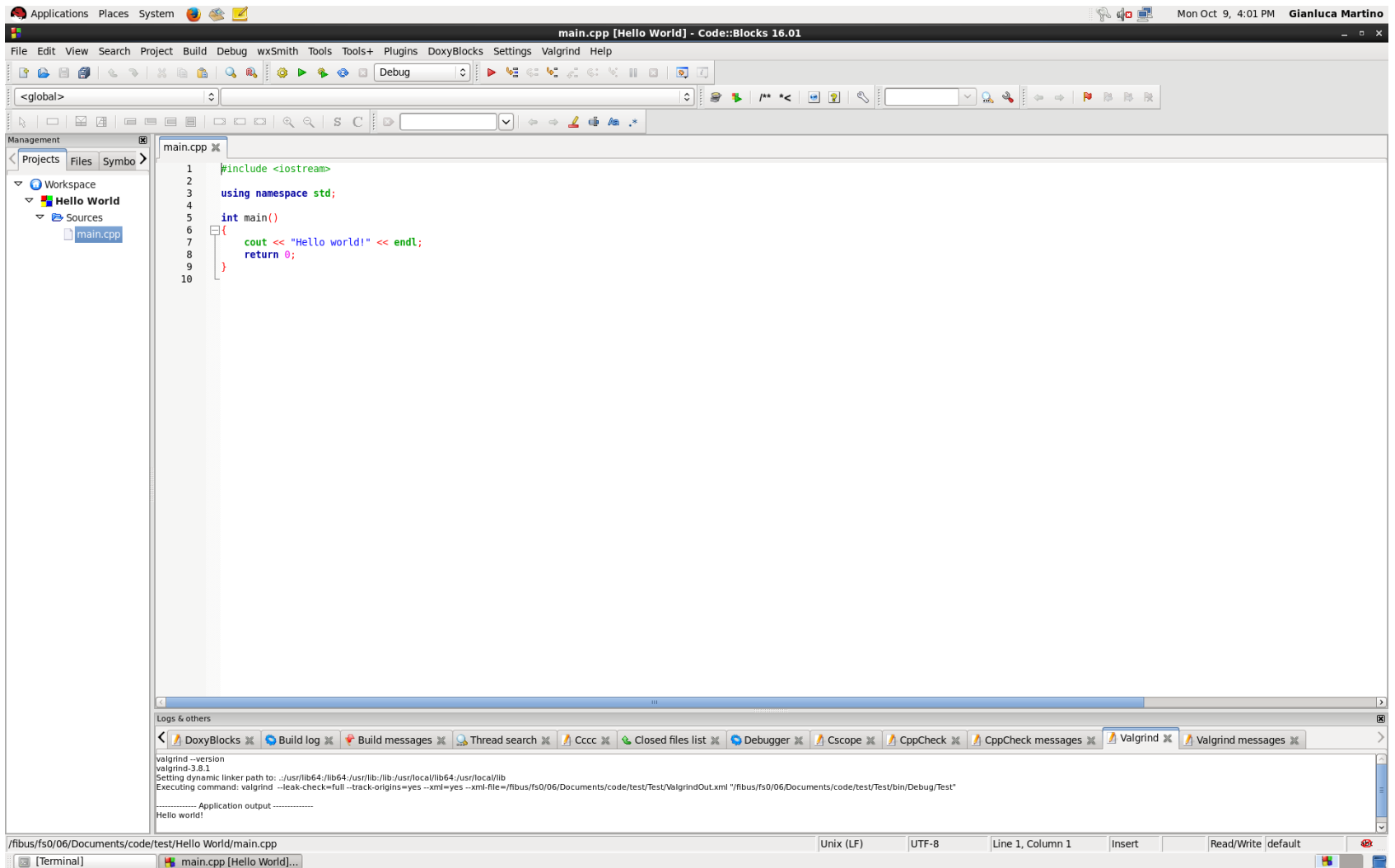
## Übung: Alles nochmal

1. CodeBlocks starten
2. File → New → Project → Console application → C++
  - Dialog abarbeiten
3. Binary erzeugen („Build Project“) durch Zahnrad klicken
4. Programm starten: Grüner Pfeil (Play)
5. Klicken auf Sources → main.cpp
6. Wieder herumspielen!

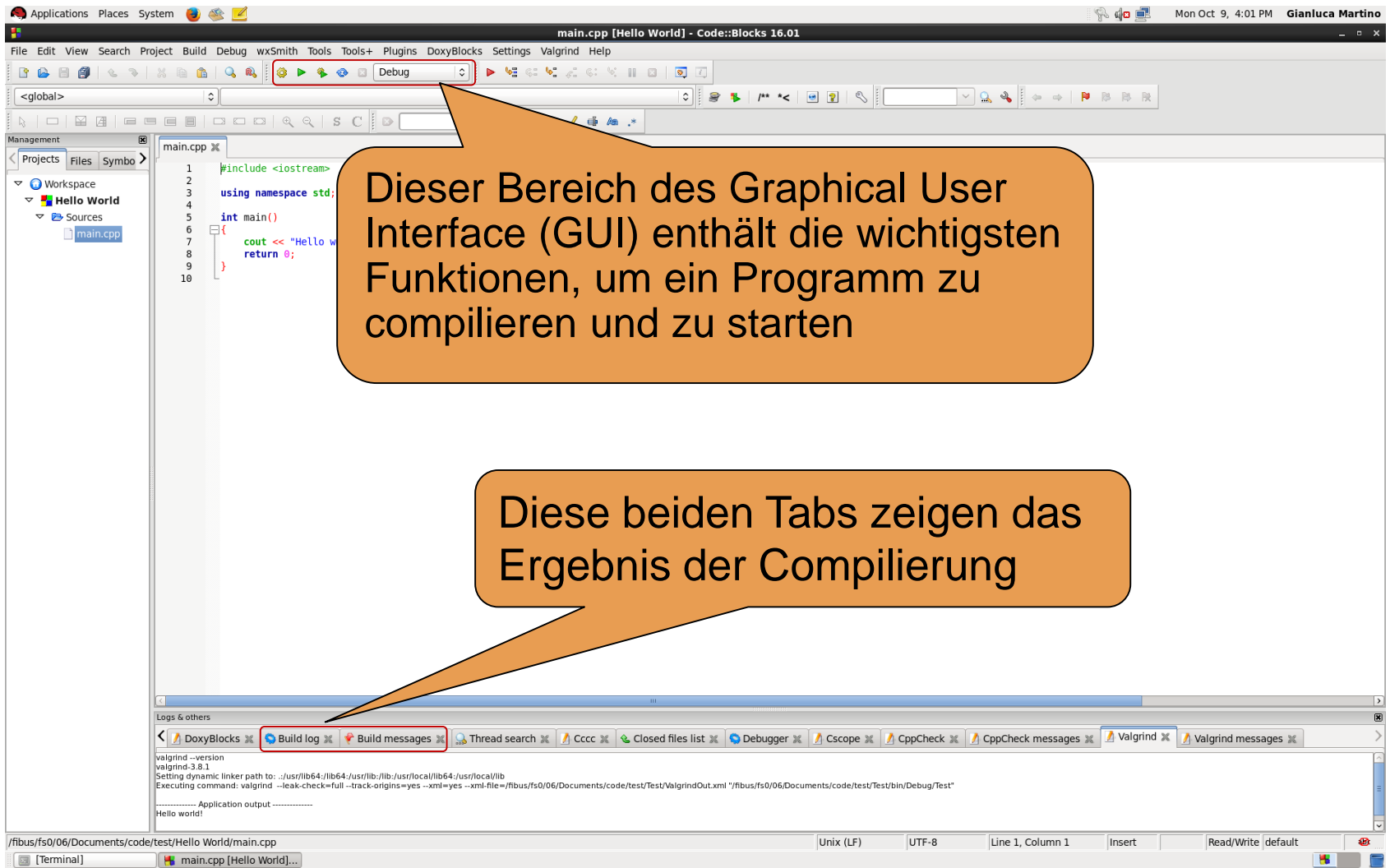
# Getting started



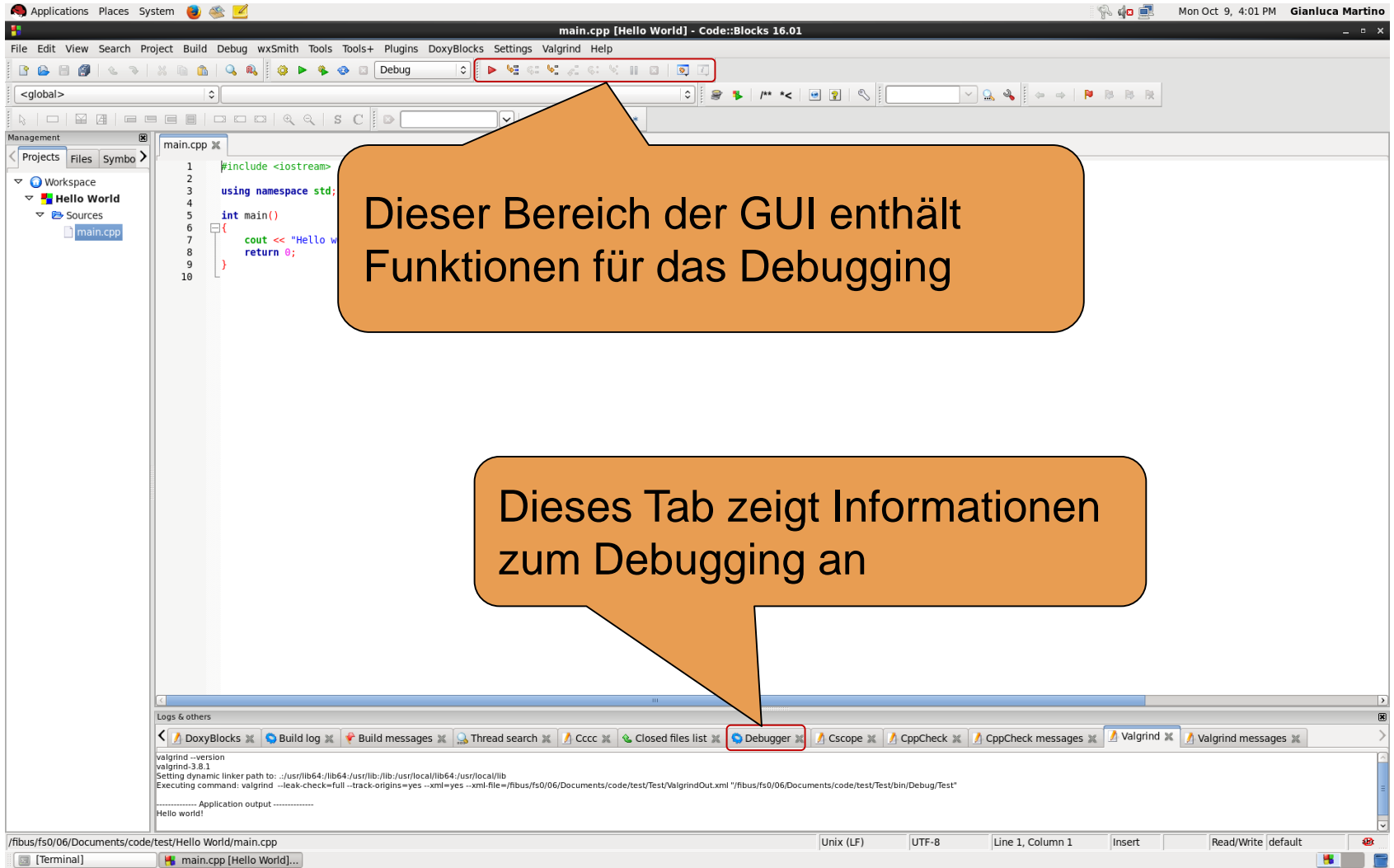
# IDE Interface



# IDE Interface



# IDE Interface



Dieser Bereich der GUI enthält Funktionen für das Debugging

Dieses Tab zeigt Informationen zum Debugging an



# IDE Interface

