

## Einleitung

Für die Bearbeitung der Übungsaufgaben der Lehrveranstaltung benötigst du Zugang zu einem Rechner mit einem C++-Compiler und einer entsprechenden Entwicklungsumgebung. Der Compiler (zusammen mit dem Linker) übersetzt den in Textform vorliegenden Programmcode in eine ausführbare Datei. Die Entwicklungsumgebung ist im Prinzip nur ein Texteditor mit einigen Zusatzfunktionen, die den Programmierer beim Erstellen des Programmcodes unterstützt und auf Knopfdruck den Compiler startet. Wir haben in den Poolräumen und für die Installation auf Windows-Laptops den *GCC-Compiler* und die *Eclipse CDT-Entwicklungsumgebung* bereitgestellt.

In diesem Dokument findest du eine Anleitung, wie du auf einem Poolrechner oder auf deinem Laptop das erste Programm kompilieren und ausführen kannst.

**Zusatzinformation** *Warum setzen wir in der Lehrveranstaltung Eclipse CDT und GCC ein?* GCC (GNU-Compiler-Collection) ist Open-Source (also frei zugänglich) und läuft auf fast allen Plattformen (Linux, Windows,...) und kann Binaries für eine noch größere Anzahl von Plattformen erzeugen – inklusive zahlreicher eingebetteter Systeme. Dieser Compiler lässt sich kostenfrei sowohl unter Linux als auch unter Windows (unter dem Namen MinGW) einsetzen. Eclipse ist ebenfalls Open-Source und generell in der Softwareentwicklung weit verbreitet. Eclipse CDT liegt eine „normales“ Eclipse für Java zugrunde, welches jedoch mit Plugins um die C++-Funktionalität erweitert wird. Im Prinzip funktionieren für die Aufgaben der Lehrveranstaltung auch beliebige andere C++-Compiler und Entwicklungsumgebungen – dafür können wir jedoch keine Unterstützung anbieten.

## 0.1 Zugang zu den Lehrmaterialien über koaLA

Die Materialien zu der Lehrveranstaltung werden online über das koaLA-System zur Verfügung gestellt. Das System erreichst du unter <https://koala.uni-paderborn.de>. Damit du in koaLA zu der Lehrveranstaltung registriert werden kannst, melde dich auf der koaLA-Seite mit deinem IMT-Login an. Nach der ersten Anmeldung wird die automatische Synchronisierung mit PAUL gestartet, so dass du **ab dem nächsten Tag** in koaLA auf die Daten zugreifen kannst.

## 0.2 Für Laptopübungen: Eigenen Laptop vorbereiten

Um für eine Übung den eigenen Laptop nutzen zu können, musst du einen C++-Compiler und eine Entwicklungsumgebung installiert haben. Im Rahmen der Übungen verwende bitte die zur Verfügung gestellte Version von MinGW und Eclipse CDT für Windows-Rechner (ab Windows XP) – andere Compiler und Umgebungen können auch funktionieren, wir können dafür aber keinerlei Unterstützung anbieten. Zur Vorbereitung lade bitte den Installer herunter (Dateigröße ca. 300MB; Installiert ca. 500MB), führe ihn aus und folge den Anweisungen:

[http://groups.uni-paderborn.de/fg-madh/ws\\_1516/GdPMB\\_v2.exe](http://groups.uni-paderborn.de/fg-madh/ws_1516/GdPMB_v2.exe)

### Zusatzinformation

- Standardmäßig wird die Software unter deinem Benutzerverzeichnis installiert; wenn du sie unter Programme installieren möchtest, kann es sein, dass du den Installer als Administrator

ausführen musst (Rechtsklick -> Als Administrator ausführen).

- Zum Deinstallieren kann der angelegte Ordner einfach gelöscht werden.
- Für den Installer wurden im Eclipse CDT einige problematische Plugins deaktiviert und es wurde ein Starter hinzugefügt, der einige Pfade anpasst – daher ist es einfacher den Installer zu benutzen als Eclipse CDT manuell zu installieren.

Mit **MacOS** lässt sich Eclipse CDT ebenfalls verwenden. Hier findest du eine entsprechende Anleitung: [http://groups.uni-paderborn.de/fg-madh/ws\\_1516/MacReadme.txt](http://groups.uni-paderborn.de/fg-madh/ws_1516/MacReadme.txt)  
Wir können jedoch keine weitere Unterstützung dafür anbieten.

### 0.3 Für Poolräume: Freischalten des Netzlaufwerks und des Windows Logins

Jeder Nutzer mit gültigem IMT-Login verfügt über einen Speicherbereich im IMT. Dieser wird benötigt um in einem Poolraum arbeiten zu können und Daten (wie z.B. die Übungsaufgaben) dauerhaft speichern zu können. Um auf dieses Laufwerk zugreifen zu können, muss jeder Nutzer seinen Bereich zunächst einmalig selber freischalten.

1. Dazu brauchst du einen Rechner mit Internetzugang. (Wenn du dich selbst nicht einloggen kannst, frage einfach einen hilfsbereiten Kommilitonen der sich bereits anmelden kann).
2. Dann öffnest du im Web-Browser den Bereich Benutzerverwaltung des IMT: <https://benutzerverwaltung.uni-paderborn.de> Auf dieser Seite wählst du „Benutzerdaten selbst verwalten“ aus und loggst dich mit deinem IMT-Login und Passwort an.
3. Wähle den Punkt „Rechnerzugang und Homepage“. Hier kannst du das Verzeichnis für private und öffentliche Daten freischalten. Wenn hier steht, wie viel Speicherplatz auf deinem Laufwerk frei ist, hast du diesen Schritt bereits erledigt.
4. Melde dich von der Seite mittels des Abmelde-Knopfes ab.
5. Warte fünf Minuten (in der Zeit kann sich der Nächste einloggen).
6. Wenn du auf einer virtuellen Maschine angemeldet bist, fahre diese herunter (Start -> Runterfahren) und starte sie neu.

Sobald du dich jetzt auf einem Windows-Rechner mit deinem IMT-Zugang eingeloggt hast, solltest du automatisch ein Laufwerk **U:** haben, das mit deinem CIF-FS-home-Verzeichnis verknüpft ist. Hier gespeicherte Daten bleiben dauerhaft erhalten und du kannst auch von deinem eigenen Rechner darauf zugreifen (siehe dazu die Hinweise beim IMT).

Weitere Hinweise findest du beim IMT unter: <https://hilfe.uni-paderborn.de/Netzwerkspeicher>  
Bei Problemen kannst du dich auch an das Notebook-café wenden: <http://nbc.uni-paderborn.de/>

## 0.4 Im Poolraum anmelden und Eclipse starten

Je nach Poolraum müsst ihr verschiedene Schritte durchführen, um Eclipse zu starten:

**Raum P1.204** Die Rechner in diesem Raum sind Terminals, die sich mit virtuellen Maschinen verbinden – d.h. die Rechner selber dienen nur zur Anzeige, eure Programme werden im Rechenzentrum der Uni ausgeführt. Meldet euch zunächst mit eurem IMT-Account am System an. Dann bekommt ihr eine Liste von möglichen virtuellen Maschinen zur Auswahl. Wählt hier die Maschine **THET-Lehrveranstaltung** aus. Bitte weiterlesen bei Abschnitt *Virtuelle Maschine*.

**Raum N5.206 und N2.216** Auf den Rechnern in diesen Räumen läuft ein normales Windows – in den Übungen arbeiten wir jedoch nicht direkt auf den Rechnern sondern auf einer virtuellen Maschine die im Rechenzentrum der Uni ausgeführt wird. Meldet euch zunächst mit eurem IMT-Account am System an. Startet dann das Programm **VMware Horizon View Client** auf dem Desktop und bestätigt so lange, bis ihr eine Auswahl von virtuellen Maschinen bekommt. Wählt hier die Maschine **THET-Lehrveranstaltung** aus.

**Virtuelle Maschine: THET-Lehrveranstaltung** In der virtuellen Maschine *THET-Lehrveranstaltung* findet ihr auf dem Desktop eine Verknüpfung mit Eclipse. Wenn ihr diese startet werdet ihr nach dem Speicherort für euren Workspace gefragt. Das ist das Verzeichnis in dem eure Projekte gespeichert werden. Standardmäßig ist euer Netzwerkspeicherplatz (siehe Abschnitt 0.3) an das Laufwerk U: gebunden; wobei Daten im Unterverzeichnis data abgelegt werden. Gebt als Pfad für euren Workspace daher `u:\data\workspace` ein.

**Raum P1.6.12.4 und P7.2.02.01** Auf den Rechnern in diesen Räumen läuft Linux. Du kannst dich auch hier mit deinem IMT-Account anmelden. Eclipse findest du im Startmenü: App -> Development (oder Entwicklung) -> Eclipse. Als Verzeichnis für dein Workspace solltest du `~/data/workspace` eintragen. Falls das nicht funktioniert, du deinen Netzwerkspeicher aber wie in Abschnitt 0.3 eingerichtet hast, probiere bitte den Pfad `~/workspace` aus.

## 0.5 Das erste Programm

Wenn Eclipse läuft und du das Workspace-Verzeichnis angegeben hast, musst du ggf. zunächst das einmalig erscheinende **Willkommensfenster in Eclipse schließen**. Dann kannst du dein erstes Projekt erstellen:

**Projekt anlegen (Abbildung 1):** Um ein „Hallo Welt“-Beispielprojekt anzulegen, wähle in Eclipse folgende Menüpunkte aus File -> New -> C++ Project. Unter Project name gibst du `Aufgabe_0` ein. (*Anmerkung:* Verwende möglichst immer einen nachvollziehbaren Name, aber **ohne Umlaute, Sonderzeichen (wie Punkt oder Komma) oder Leerzeichen!** Unterstriche `_` sind erlaubt.) Im erscheinenden Fenster wähle unter Project Type die Option Executable / Hello World C++ Project aus. Im Rechten Teil Toolchains wähle MinGW GCC (oder Linux GCC) aus. Falls du gefragt wirst, ob Eclipse in die C++-Perspektive wechseln soll, bestätige das.

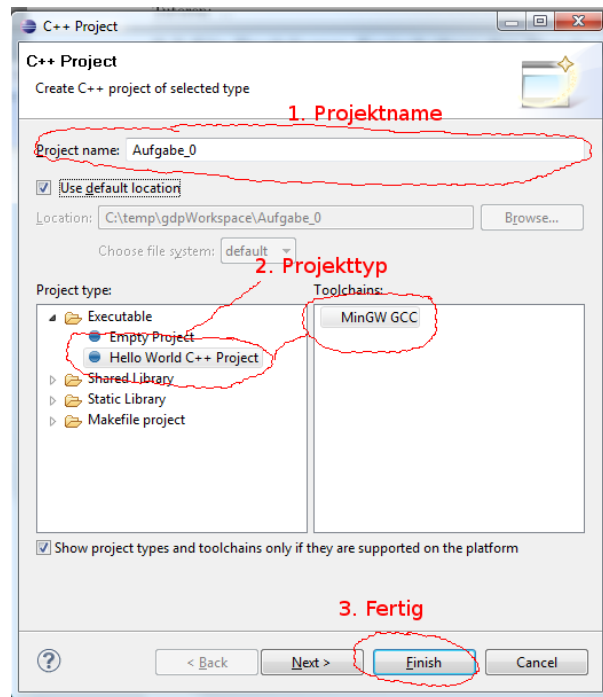


Abbildung 1: Erzeuge neues Projekt (File -> New -> C++ Project)

**Hinweis** Sollten die genannten Optionen nicht verfügbar sein, versuche die Option „Show projects types and toolchains only if they are supported by the plattform“ zu deaktivieren.

**Programm kompilieren:** Nach der Erzeugung des neuen Projektes, solltest du eine Projektansicht wie in Abbildung 2 erhalten. Das vorgefertigte Programm schreibt `!!!Hello World!!!` auf den Bildschirm (die klassische Ausgabe fast aller ersten Programme). Bevor das Programm ausgeführt werden kann, muss es erst vom Quellcode (Text) in ein ausführbares Binärprogramm übersetzt werden (d.h. das Programm wird kompiliert und gelinkt). Dies könnte z.B. darüber erfolgen, dass man die entsprechenden Compiler-Befehle über die Kommandozeile eingibt. Eclipse unterstützt den Entwickler jedoch bei diesem Vorgang, indem es die Befehle automatisch ausführt, sobald man auf Build klickt (das Hammer-symbol in der Symbolleiste; bei Änderungen am Quelltext das Speichern vorher nicht vergessen!). In der Ausgabeanzeige (Console-Tab am unteren Rand) kann man die ausgeführten Befehle sehen. Wenn beim Kompilieren ein Fehler auftritt, kann man diese entweder in der Ausgabeanzeige, im Problems-Tab oder durch die Annotationen direkt im Quelltext erkennen. Sollte die Erstellung des ersten Programms gelungen sein, steht in der Ausgabe etwas wie: `16:37:53 Build Finished (took 564ms)` und im Projektverzeichnis in deinem Workspace liegt im Unterordner Debug die neue ausführbare Datei `Aufgabe_0.exe`.

**Programm ausführen:** Um das Programm zu starten, muss Eclipse zunächst wissen, welche Datei dafür ausgeführt werden muss. Dazu öffne (nachdem die Datei erfolgreich kompiliert wurde) im Projektbaum den Punkt Binaries und klicke dein erzeugtes Programm (z.B. `Aufgabe_0.exe`) mit einem Rechtsklick an (siehe Abbildung 3). In dem sich öffnenden Menü, wähle `Run As -> Local C/C++ Application` aus. Nach kurzer Zeit sollte die Ausgabe des Programms im Ausgabefenster erscheinen: `!!!Hello World!!!`

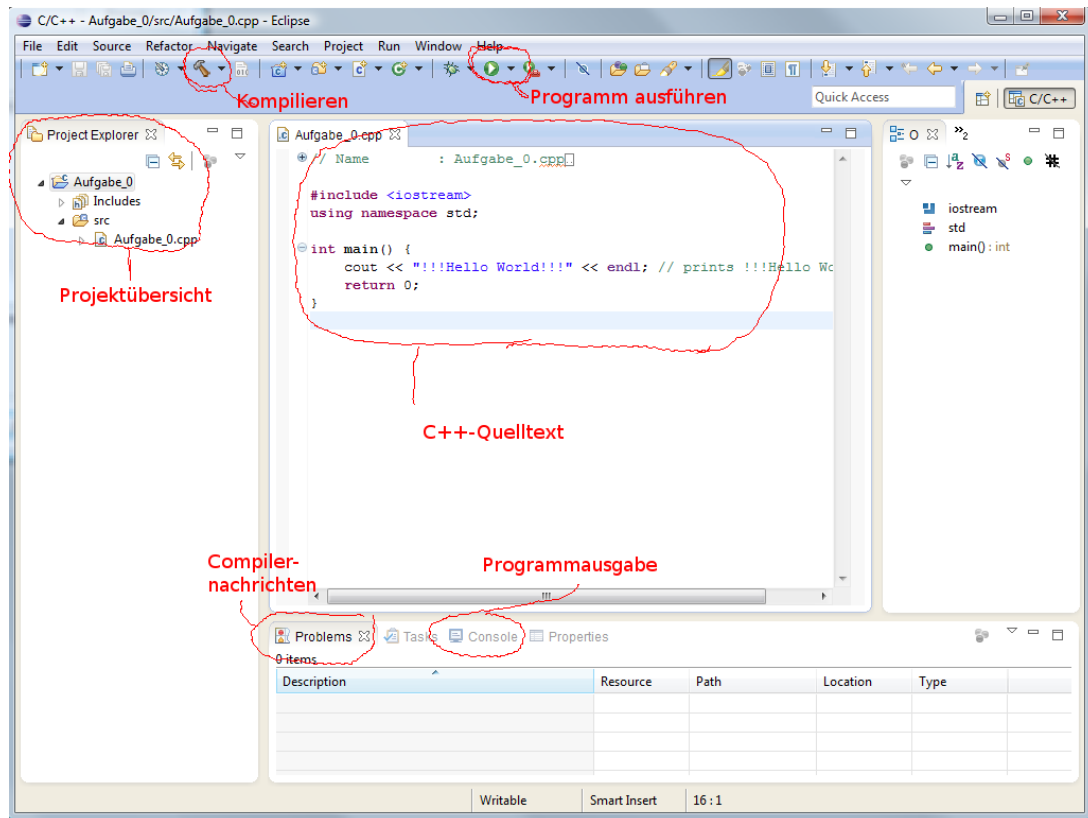


Abbildung 2: Projektansicht

Nachdem du damit Eclipse einmal gezeigt hast, welche die ausführbare Datei ist, solltest du in Zukunft das Programm auch mit dem Run-Button starten (weißes Dreieck in grünem Kreis in der Toolbar).

## 0.6 Aufgabe

Ändere das Programm so, dass es Hallo Paderborn! ausgibt.

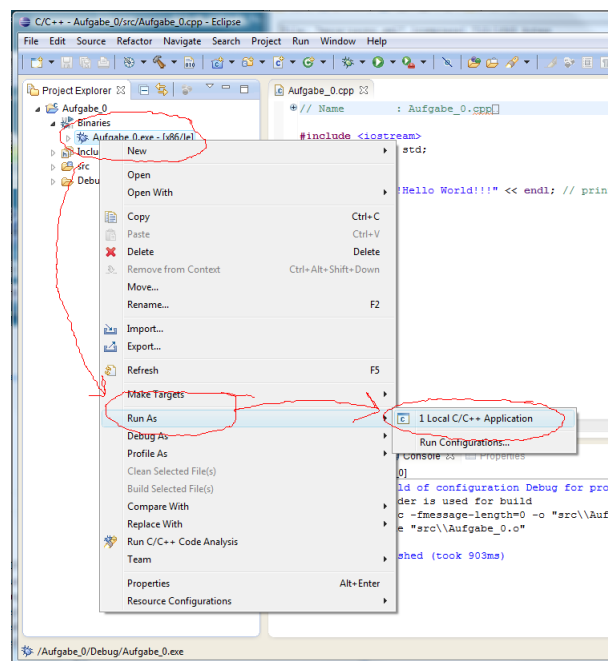


Abbildung 3: Das fertige Programm starten

## 0.7 Weitere Hinweise zur Nutzung von Eclipse

- Projektnamen sollten nur aus Buchstaben, Zahlen und Unterstrichen (\_) bestehen. Andere Zeichen (wie Punkte oder Leerzeichen) können Probleme bereiten.
- Um ein Projekt zu löschen, wähle es im Projektbaum mit der rechten Maustaste aus und wähle Delete. Wähle dabei aus, dass auch der Inhalt des Projektes gelöscht werden soll. Andernfalls bleiben die eigentlichen Dateien in deinem Workspaceverzeichnis erhalten und du musst sie ggf. von Hand im Dateimanager löschen.
- Wenn das Fenster von Eclipse nicht so aussieht wie es sollte (es fehlt z.B. der Projektbaum), dann versuche: Eclipse neu starten und dann im Menü Window -> Show View -> C/C++ Projects. Es kann auch sein, dass die C++-Perspektive nicht aktiviert ist. Um das zu korrigieren, wähle im Menü Window -> Open Perspective -> Other... -> C/C++ -> OK. Wenn das nicht hilft, kann auch Window -> Reset Perspective helfen.
- Falls beim Kompilieren keine ausführbare Datei erstellt werden kann, dann kann es helfen in den Optionen neben dem Kompilier-Button (Hammersymbol) die Funktion „Release“ auszuwählen.
- Auf dem eigenen Laptop kann scheinbar auch der Virenschanner das Kompilieren verhindern. Dann kann es helfen den Virenschanner zwischenzeitlich zu deaktivieren.
- Falls das selbst geschriebene Programm nicht richtig beendet wurde, kann beim Kompilieren folgende Fehlermeldung in der Konsole erscheinen (in etwa):  

```
../mingw32/bin/ld.exe: cannot open output file filename.exe: Permission denied  
collect2.exe: error: ld returned 1 exit status
```

In diesem Falle kann man das noch laufende Programm folgendermaßen beenden: Oben rechts im Console-Tab die Konsolenliste ausklappen (blaues Monitor Icon „Display Selected Console“) -> Nacheinander alle Einträge mit [C/C++ Application], aber ohne <terminated> auswählen -> für jeden Eintrag im Console-Tab das Programm mit dem roten Terminate-Icon beenden. Wenn hinter allen Einträgen in der Konsolenliste <terminated> steht, sollten alle laufenden Instanzen beendet sein und das Programm sollte sich jetzt wieder kompilieren lassen.
- Wenn man das fertig kompilierte Programm von Hand aus dem Dateimanager oder aus der Kommandozeile startet, kann eine Fehlermeldung “???.dll nicht gefunden” erscheinen. Das Programm verwendet Befehle, die in einer externen Bibliothek (eben der nicht gefundenen DLL) implementiert sind. Wenn das Programm von Eclipse aus gestartet wird, liegen diese Bibliotheken im Pfad und können vom Betriebssystem gefunden werden. Kopiere einfach die jeweils geforderten DLL-Dateien aus dem MinGW\bin (der Ort kann leider je nach Installation unterschiedlich sein) in das Verzeichnis deines Programms, damit sie gefunden werden.
- Wenn man das Programm aus dem Dateimanager startet, geht nur kurz ein Fenster auf und sofort wieder zu. Das liegt daran, dass das Programm lediglich etwas auf die Kommandozeile ausgibt und nicht auf Benutzereingaben wartet. Daher beendet es sich nach der Ausgabe sofort wieder. Um die Ausgabe auch ohne Eclipse zu sehen, starte das Programm über die Kommandozeile (unter Windows z.B. cmd.exe).
- Weitere Hinweise folgen bei Bedarf...