

# Mein optimiertes Dokument

6. August 2023

## 1 Programmiersprachen

Aufgaben zum C/C++ Kurs / Universität Regensburg / Fakultät Physik

# ChatGPT

Aufgabe: " "

Entwickeln Sie ein Programm in den Programmiersprachen C, C++, Python,  
↪ Arduino und PHP.

Das Programm soll: Bei fehlerhaften Eingaben (z.B. wenn der Benutzer keine  
↪ Zahl eingibt)  
sollte das Programm den Benutzer darauf hinweisen und ihn erneut zur  
↪ Eingabe auffordern.

Anforderungen:

- Implementieren Sie Fehlerprüfungen für die Eingabe.
- Kommentieren Sie den Code auf Deutsch, wo es sinnvoll und notwendig ist.
- Befolgen Sie die Best Practices

Nachdem Sie das Programm in allen fünf Programmiersprachen erstellt haben,  
dokumentieren Sie die Testergebnisse in Markdown.

## 2 Code auszuführen

### 2.0.1 C:

1. Installieren Sie einen C-Compiler, z. B. GCC.
2. Speichern Sie den Code in einer Datei namens `hello.c`.
3. Öffnen Sie das Terminal oder die Kommandozeile.
4. Kompilieren Sie den Code mit: `gcc hello.c -o hello`
5. Führen Sie das Programm aus: `./hello` (oder `hello.exe` auf Windows).

### 2.0.2 C++:

1. Installieren Sie einen C++-Compiler, z. B. G++.
2. Speichern Sie den Code in einer Datei namens `hello.cpp`.
3. Öffnen Sie das Terminal oder die Kommandozeile.

4. Kompilieren Sie den Code mit: `g++ hello.cpp -o hello`
5. Führen Sie das Programm aus: `./hello` (oder `hello.exe` auf Windows).

### 2.0.3 Python:

1. Installieren Sie Python von [python.org](https://python.org).
2. Speichern Sie den Code in einer Datei namens `hello.py`.
3. Öffnen Sie das Terminal oder die Kommandozeile.
4. Führen Sie das Skript aus: `python hello.py`

### 2.0.4 Arduino:

1. Installieren Sie die Arduino IDE von der [Arduino-Website](https://www.arduino.cc).
2. Öffnen Sie die Arduino IDE und kopieren Sie den Code in das Textfeld.
3. Verbinden Sie Ihren Arduino über USB.
4. Wählen Sie unter **Werkzeuge** -> **Port** den entsprechenden COM-Port aus.
5. Klicken Sie auf **Hochladen**.
6. Öffnen Sie den Seriellen Monitor, um Eingaben zu tätigen und Ausgaben zu sehen.

### 2.0.5 PHP:

1. Installieren Sie PHP von [php.net](https://php.net).
2. Speichern Sie den Code in einer Datei namens `hello.php`.
3. Öffnen Sie das Terminal oder die Kommandozeile.
4. Führen Sie das Skript aus: `php hello.php`

## 3 Makefile

Ein **Makefile** ist ein nützliches Tool, um Build-Prozesse zu automatisieren. Es ist besonders hilfreich, wenn Sie mehrere Dateien haben oder Abhängigkeiten zwischen diesen Dateien bestehen. Für den bereitgestellten Code (eine einfache `hello.c` für C und `hello.cpp` für C++) kann ein **Makefile** recht einfach sein.

Hier ist ein **Makefile**, das sowohl für den C- als auch für den C++-Code verwendet werden kann:

```
# Compiler und Flags definieren
CC = gcc
CXX = g++
CFLAGS = -std=c17 -Wall -O2
CXXFLAGS = -std=c++20 -Wall -O2
LDLAGS =

# Ziel- und Quell-Dateien definieren
C_SOURCES = main.c
CPP_SOURCES = main.cpp
C_OBJECTS = $(C_SOURCES:.c=.o)
CPP_OBJECTS = $(CPP_SOURCES:.cpp=.o)
TARGET_C = main_c
TARGET_CPP = main_cpp

# Regel, um das C-Programm zu erstellen
$(TARGET_C): $(C_OBJECTS)
```

```

$(CC) $(C_OBJECTS) -o $(TARGET_C) $(LDFLAGS)

# Regel, um das C++-Programm zu erstellen
$(TARGET_CPP): $(CPP_OBJECTS)
    $(CXX) $(CPP_OBJECTS) -o $(TARGET_CPP) $(LDFLAGS)

# Regel, um C-Objekt-Dateien zu erstellen
%.o: %.c
    $(CC) -c $(CFLAGS) $< -o $@

# Regel, um C++-Objekt-Dateien zu erstellen
%.o: %.cpp
    $(CXX) -c $(CXXFLAGS) $< -o $@

# Zusätzliche nützliche Regeln
clean:
    rm -f $(C_OBJECTS) $(CPP_OBJECTS) $(TARGET_C) $(TARGET_CPP)

all: $(TARGET_C) $(TARGET_CPP)

```

Um die Programme zu kompilieren, können Sie folgende Befehle verwenden:

- Für das C-Programm: `make $(TARGET\_C)`
- Für das C++-Programm: `make $(TARGET\_CPP)`
- Für beide Programme: `make all`
- Zum Bereinigen der kompilierten Dateien: `make clean`

Stellen Sie sicher, dass Sie die Variablen `C\_SOURCES`, `CPP\_SOURCES`, `TARGET\_C` und `TARGET`  $\rightarrow$  `\_CPP` an Ihren eigenen Code anpassen.