

Computergrundlagen 2025

Blatt 14: C++

- Abgabetermin für die Lösungen: **01.02.2026, 20 Uhr/ für Montagsgruppe: 30.01.2026, 12 Uhr**
- Bei Fragen wendet euch bitte an eure/n Tutor/in:
 - Mo 11:30: Stephan Haag: `st170833@stud.uni-stuttgart.de`
 - Di 09:45: Julian Hoßbach: `julian.hossbach@icp.uni-stuttgart.de`
 - Mi 14:00: Julian Peters: `julian.peters@icp.uni-stuttgart.de`
 - Do 09:45: Rebecca Stephan: `rebecca.stephan@icp.uni-stuttgart.de`
 - Fr 09:45: Jonas Höpker: `st182335@stud.uni-stuttgart.de`
- Die Übungsaufgaben sollen in der Regel in **Zweiergruppen** bearbeitet werden. Nur in **begründeten Ausnahmefällen** sind Dreiergruppen möglich.
- Die Abgabe der Übungsblätter erfolgt über Ilias.
- Mit Abgabe der Lösungen erklärt Ihr, dass Ihr die Lösung euren Mitstudierenden im Rahmen der Übungsbesprechung vorstellen könnt. Um dies zu überprüfen, muss mindestens zweimal von jedem Teilnehmenden vorgetragen werden. Wenn Ihr das nicht könnt, werden euch die Punkte für die entsprechenden Aufgaben wieder abgezogen.
- **Befehle, die nicht in der Vorlesung besprochen wurden, müssen gegebenenfalls recherchiert werden.**
- **Alle erstellen Skripte sowie ein mit markdown oder Latex erstellter Report (.pdf) sind Teil der Abgabe**

Sortieralgorithmen in C++ (2 Punkte)

Ergänze folgenden Code so, dass die Personen Alice, Bob und Clara nach Alter sortiert ausgegeben werden (siehe `main` Funktion).

Schreibe dazu eine Funktion `is_younger` und verwende diese mit `std::sort`.

Die Stellen, die du ergänzen sollst sind mit `TODO` gekennzeichnet

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

class Person {
public:
    std::string name;
    int age;
};

bool is_younger(const Person& a, const Person& b) {
    TODO
}

int main() {
    std::vector<Person> people{
        {"Alice", 30},
```

```

        {"Bob", 24},
        {"Clara", 41}
    };
    // std::sort mit der Funktion is_younger aufrufen
    TODO

    // Ausgabe zur Überprüfung
    TODO
}

```

Klassen in C++ (3 Punkte)

Erstelle eine Klasse `Particle`, die physikalische Eigenschaften eines Teilchens in einem 2D-Raum abbildet.

Die Klasse soll folgende Member-Variablen (Datentypen nach eigenem Ermessen) besitzen: `x`, `y` (Position), `vx`, `vy` (Geschwindigkeit) und `mass` (Masse).

Implementiere die Member-Funktionen `momentum_x()` und `momentum_y()`. Diese sollen den Impuls in der jeweiligen Richtung berechnen und zurückgeben ($p = m \cdot v$).

Schreibe eine `main()`-Funktion, in der ein Objekt der Klasse `Particle` instanziiert wird. Weise Werte zu und gib den berechneten Impuls (in x - und in y -Richtung) auf der Konsole aus.

Git Grundlagen (5 Punkte)

In dieser Übung lernst du den Standard-Workflow kennen, wie Software weltweit kollaborativ entwickelt wird. Du wirst ein Projekt kopieren (forken), lokal bearbeiten und deine Änderungen zur Korrektur einreichen.

1. Vorbereitung (Einmalig)

Überprüfe ob auf deinem Computer `git` bereits installiert ist. Öffne dafür ein Terminal gebe `git --version` ein. Auf den Computern im CIP-Pool ist `git` bereits installiert. Falls `git` installiert ist `git`, installiere es. Das Vorgehen unterscheidet sich dabei zwischen den Betriebssystemen. Sobald `git` installiert ist musst du deine Identität setzen. `Git` muss wissen, wer du bist. Tippe im Terminal:

```

git config --global user.name "Dein Vorname Nachname"

git config --global user.email "deine.email@beispiel.de"

```

2. Das Repository "Forken"

Gehe auf die Webseite des Repositories für diese Aufgabe: <https://github.com/tobias-flnr/cgl-git-uebung>.

Stelle sicher, dass du bei GitHub eingeloggt bist. Falls du noch keinen Account hast, erstelle einen. Du wirst ihn sehr wahrscheinlich noch oft in deinem Leben brauchen.

Auf der Seite des Repositories, klicke oben rechts auf den Button "Fork". Wähle deinen Account aus. Du hast nun eine exakte Kopie des Projekts unter deinem eigenen Profil (der

Link sollte in etwa so ausschauen: `github.com/dein-name/cgl-gitgit-uebung`).

Hinweis: Falls du mit VS Code arbeitest kannst du auch die `git` Benutzeroberfläche verwenden. Eine Anleitung dazu findest du hier: <https://code.visualstudio.com/docs/sourcecontrol/overview>

3. Lokale Kopie erstellen (Clone)

Kopiere nun deinen Fork auf deinen Computer: Klicke in deinem Fork auf den grünen Button “Code” und kopiere die HTTPS-URL.

Öffne dein Terminal / Eingabeaufforderung und tippe: `git clone [DEINE_KOPIERTE_URL]`

Wechsle in den Ordner: `cd cgl-gitgit-uebung`

4. Einen neuen Zweig anlegen (Branch)

Es ist eine gute Angewohnheit nicht direkt im Hauptzweig (main). Erstelle einen neuen Zweig für dein Feature:

```
git checkout -b neues_feature
```

5. Code bearbeiten & Speichern

Öffne die Datei `hallo.txt` in einem Editor.

Ändere den Text von “Hallo” zu “Hallo Welt”.

Speichere die Datei.

Füge die Änderung hinzu (“stagen”): `git add hallo.txt`

Erstelle einen Speicherpunkt (Commit): `git commit -m "Begrüßung angepasst"`

6. Änderungen hochladen (Push)

Schicke deinen neuen Zweig zurück auf den Server in deinen Fork:

```
git push origin neues_feature
```

(Hinweis: Eventuell musst du dich hier im Browser oder via Token anmelden).

7. Pull Request (PR) & Abgabe

Gehe im Browser auf die Seite deines Forks. Dort erscheint ein gelber Balken mit der Aufschrift “Compare & pull request”. Klicke darauf. Prüfe, ob dein neuer Zweig (`neues_feature`) in das Original-Repo (`main`) geschickt werden soll. Klicke auf “Create pull request” und erstelle die pull request um den branch `neues_feature` in den `main` Zweig deines Forks des Repositories zu mergen.

Abgabe: Kopiere den Link deines öffentlichen Repositories und reiche ihn zusammen mit dem Link zu deinem Pull Request ein. Achte dabei, dass dein Fork öffentlich ist, sodass deine Tutorin oder dein Tutor zugreifen kann. Dokumentiere für die Abgabe zudem alle wichtigen schritte und Probleme, die möglicherweise aufgetreten sind.