

## 2. Übung: Einfache Datentypen und Schleifen

**Ziel:** Umgang mit arithmetischen Datentypen und Schleifen.

---

### Vorbereitung zu Hause

- Entwurf der Programme
- Quellen erstellen mit irgendeinem Editor (Notepad, vi, keine C++-Entwicklungsumgebung erforderlich!)
- eventuell Syntaxcheck und erste Tests mit g++

*gilt auch für alle künftigen  
Übungen*

---

### Allgemeines für das Praktikum

- Legen Sie mit NetBeans einen eigenen Arbeitsbereich für jede Übung an und fügen Sie für jede Aufgabe ein C++ Projekt zum Arbeitsbereich hinzu.
- Falls Ihr Programm nicht das tut, was es soll: verwenden Sie den Debugger, oder eigene Testausgaben.

Achten Sie bei **allen** Aufgaben (auch in den künftigen Übungen) auf:

- aussagekräftige Namen
- Dokumentation (Programmkopf) und Kommentierung
- Ausgabe eines "Ansagetextes" zu Beginn des Programms
- selbsterklärende Eingabedialoge
- sinnvolle Auswahl der verwendeten Datentypen
- saubere Strukturierung des Programmtextes durch Einrückung

---

### 1. Aufgabe: Gerade Zahlen

Schreiben Sie ein Programm, das alle **geraden Zahlen** in einem bestimmten Zahlenbereich ausgibt. Der Zahlenbereich ist durch Konstanten festzulegen.

**Tipp:** verwenden Sie den Modulo-Operator %, um gerade und ungerade Zahlen zu unterscheiden

---

### 2. Aufgabe: Verzinsung

Schreiben Sie ein Programm zur Berechnung des Wertzuwachses eines **festverzinslichen Wertpapiers** bei jährlicher Verzinsung. Das Programm soll vom Benutzer das Anfangskapital, den Zinssatz in Prozent und die Laufzeit in Jahren erfragen. Es gibt dann für jedes Jahr der Laufzeit den aktuellen Wert des Wertpapiers aus.

**Tipp:** die Formel zur Zinsberechnung:  $\text{Kapital}_{\text{Jahr}+1} = \text{Kapital}_{\text{Jahr}} * (1 + \text{Zinssatz}/100)$

Benutzen Sie nur die Grundrechenarten, keine Bibliotheksfunktionen!

---

### 3. Aufgabe: Primzahlen

Schreiben Sie ein Programm, das alle **Primzahlen** in einem bestimmten Zahlenbereich berechnet und ausgibt. Der Zahlenbereich soll nach Programmstart vom Benutzer abgefragt werden (nur positive ganze Zahlen).

**Tipp:** verwenden Sie den Modulo-Operator %

---

### 4. Aufgabe: Datenanalyse

Schreiben Sie ein Programm, das eine Folge von nicht negativen Zahlen einliest und das Minimum, das Maximum und den Mittelwert davon berechnet.