

## Curriculum

# Programmieren in C++

Modulkürzel: prcpp

Modulname: Programmieren in C++

Zugehörige Modulbeschreibung:

Verfasser: C. Stamm

Datum: 19.7.2024

# 1. Funktion im Rahmen der Gesamtausbildung

## 1.1. Leitidee

Mit der Verabschiedung der modernen Standards (ab C++11) hat C++ wieder vermehrt Aufmerksamkeit bei den grossen Software-Herstellern erlangt. Nicht zuletzt die mobilen Gerätehersteller haben erkannt, dass ressourceneffiziente Programmierung nach wie vor entscheidend ist und dass C++ alle Voraussetzungen erfüllt, um sehr effiziente, skalierbare und stabile Software zu entwickeln. Der neue Standard C++20 räumt mit ein paar alten Unschönheiten auf und bringt einen bunten Strauss an Neuerungen – inspiriert durch C# und Java – mit sich. Wir sind überzeugt, dass die Industrie in den nächsten Jahren wieder vermehrt nach C++-Programmierern Ausschau halten wird. In diesem Modul praktizieren wir konsequent den neuen Standard und zeigen vor allem die grossen Unterschiede zur Java-Welt auf.

## 1.2. Voraussetzungen / Vorkenntnisse

Es werden Java-Kenntnisse vorausgesetzt, so wie sie in den Modulen Objektorientiertes Programmieren 1 und 2 angeboten werden.

# 2. Unterrichtsorganisation

## 2.1 Lern- und Arbeitsformen

Im Stundenplan sind 3 Wochenlektionen für diesen Kurs reserviert. Diese drei Lektionen werden für konzentrierte Wissensvermittlung, aber auch für kleine integrierte Übungen und interaktiven Wissensabgleich verwendet. Zu den besprochenen Themen werden passende Arbeitsblätter angeboten, welche von den Studierenden im begleiteten Selbststudium gelöst werden. Unklarheiten und Schwierigkeiten sollen von den Studierenden im wöchentlichen seminaristischen Unterricht eingebracht und zusammen mit dem Dozenten thematisiert werden.

Integraler Bestandteil des Moduls sind drei grössere Übungen, welche von den Studierenden im Selbststudium in Einzelarbeit bearbeitet werden. Diese Übungen (nur Quellcode und Resultate) müssen bis zum Unterrichtsbeginn der angegebenen Woche elektronisch dem Dozenten abgegeben werden. Die eingereichten Übungen werden begutachtet und mit einem Feedback versehen. Nur eigenständig entwickelte Lösungen werden fürs Testat angerechnet.

## 2.2. Zeitaufwand für Studierende (summarisch)

Kontaktunterricht (Vorlesung, Arbeitsblätter, Programmieren):	43 h
Begleitetes Selbststudium (Arbeitsblätter und Übungen):	32 h
Unbegleitetes Selbststudium (individuell):	15 h
Gesamtaufwand:	90h

## 2.3. Leistungsbeurteilung

### 2.3.1 Testatpflicht

Für einen erfolgreichen Abschluss des Moduls ist die Akzeptanz der drei Testatübungen durch den Dozenten notwendig. Um Akzeptanz zu erlangen, sind

1. alle zur Verfügung gestellten Unit-Tests erfolgreich zu bestehen,
2. alle Quellcode-Dateien (und allfällige weitere verlangte Unterlagen) der Testatübungen dem Dozenten per E-Mail-Anhang (unkomprimiert, keine Archivdatei) rechtzeitig abzugeben (siehe Zeitplan),
3. vom Dozenten verlangte Verbesserungen auszuführen.

### 2.3.2 Leistungsbeurteilung im Semester

Die Modulschlussnote wird durch zwei gleichberechtigte, schriftliche, angekündigte Prüfungen von je 90 Min ermittelt. In diesen Prüfungen werden sowohl Verständnisfragen als auch Programmierfragen gestellt, die auch Bezug auf die Arbeitsblätter und Übungen nehmen. Für die erste Prüfung sind 2 selbst verfasste A4-Seiten und für die zweite Prüfung insgesamt 4 selbst verfasste A4-Seiten erlaubt.

### 2.3.3 Modulschlussprüfung

Es findet keine abgesetzte Modulschlussprüfung statt.

## **2.4. Arbeitsmittel**

- verschiedene Lehrbücher
- Folien und Zusammenfassungen
- Arbeitsblätter mit Lernaufgaben
- 3 testatpflichtige Übungen
- Notebook mit C++20-Compiler (Unterrichtsbeispiele werden mit Visual C++ entwickelt)

## **3. Themenübersicht**

Das Grundthema ist der Aufbau und die Entwicklung von praxistauglichen C++20-Anwendungen. Unterschiede zu Java und typische Problemstellungen im Zusammenhang mit der Entwicklung von C++-Anwendungen werden thematisiert.

### **3.1 Einführung**

In einer ersten Einführungswoche werden die grundsätzlichen Unterschiede zwischen Java und C++ im Entwicklungsprozess thematisiert.

### **3.2 Grundlagen von C++**

Der Aufbau und die Struktur von einfachen C++-Programmen mit einfachen Datentypen, Zeigern und Referenzen, eindimensionale Arrays, C-Strings, lokalen und globalen Variablen und Funktionen mit Parameterübergabe wird anhand verschiedener kleiner Konsolenprogramme aufgezeigt. Stack, Heap, die Allokation und Freigabe von Speicher (einfache Variablen und Objekte) und Performanceunterschiede werden diskutiert.

Im Zentrum der ersten grossen Übung steht die Entwicklung/Erweiterung einer einfachen Konsolenapplikation.

### **3.3 Klassen**

Im zweiten Teil wird der Umgang mit C++-Klassen in seiner ganzen Vielfalt (Vererbung, Mehrfachvererbung, Schnittstellen, Polymorphie, abstrakte Methoden/Klassen, Methoden ohne Schreibrechte, Überladen von Operatoren) eingehend behandelt.

Die zweite grosse Übung thematisiert die Umsetzung eines abstrakten Datentyps in Form einer C++-Klasse, welche möglichst viele der zuvor genannten Konzepte umsetzt.

### **3.2 Templates**

Im dritten Teil stehen der Umgang mit Streams, der Einsatz der Standard-Bibliothek und somit auch die generische Programmierung mit Klassen- und Funktions-Templates im Zentrum. Angrenzende Themen wie Namensräume und Exceptions werden ebenfalls kurz angesprochen.

Die dritte grosse Übung beinhaltet die Programmierung mit Templates und den Einsatz der Standard-Bibliothek.

#### 4. Drehbuch (Abweichungen vorbehalten)

KW	Datum	Kontaktstudium (KS)	Selbststudium (SS)	KS	SS	Tot
38	17.09.2024 19.09.2024	Einführung von C zu C++	Arbeitsblatt 0 (Visual Studio Tutorial)	3	1	4
39	24.09.2024 26.09.2024	C++ Grundlagen Referenzen	Arbeitsblatt 1	3	1	4
40	01.10.2024 03.10.2024	Zeiger, Arrays, Zeigerarithmetik	Arbeitsblatt 2	3	1	4
41	08.10.2024 10.10.2024	strukturierte Datentypen, Klassen	Arbeitsblatt 3	3	4	7
42	15.10.2024 17.10.2024	Performance-Betrachtungen Move-Semantik	Arbeitsblatt 4	3	4	7
43	22.10.2024 24.10.2024	Operatoren überladen	Arbeitsblatt 5 <b>Abgabe Übung 1</b>	3	4	7
44	29.10.2024 31.10.2024	Vererbung, RTTI Mehrfachvererbung	Arbeitsblatt 6	3	4	7
45	05.11.2024 07.11.2024	<b>Prüfung 1</b>	Prüfungsvorbereitung	2	7	9
46	12.11.2024 14.11.2024	Prüfungsnachbesprechung		3	1	4
47	19.11.2024 21.11.2024	Templates and Variadic Templates	Arbeitsblatt 7 <b>Abgabe Übung 2</b>	3	1	4
48	26.11.2024 28.11.2024	<b>Projektwoche</b>		0	0	0
49	03.12.2024 05.12.2024	Metaprogramming, Concepts	Arbeitsblatt 8	3	1	4
50	10.12.2024 12.12.2024	Streams, Stream-Manipulatoren	Arbeitsblatt 9	3	4	7
51	17.12.2024 19.12.2024	Standardbibliothek: Container, Iteratoren und Algorithmen	Arbeitsblatt 10	3	4	7
02	07.01.2025 09.01.2025	Einsatz der Standardbibliothek und funktionale Programmierung	Arbeitsblatt 11	3	4	7
03	14.01.2025 16.01.2025	<b>Prüfung 2</b>	<b>Abgabe Übung 3</b> Prüfungsvorbereitung	2	6	8
		<b>Total</b>		<b>43</b>	<b>47</b>	<b>90</b>