

# Wahlpflichtfach

## „Die verschiedenen Programmierparadigmen von C++“ im WS 2025/26

Im Folgenden sind kurz die Randbedingungen der Veranstaltung „Die verschiedenen Programmierparadigmen von C++“ im WS 2025/26 dargestellt:

Abweichend von der Default-Prüfungsordnung greift der Dozent auf das Vorleistungspunktemodell zurück (30% der Punkte können bereits vor Klausurbeginn erreicht werden).

1. Die Vorlesung beginnt am Freitag, den 26. September 2025 mit einem Doppelblock im ersten und zweiten Block, ab 8:15 bis 11:30 vermutlich im Raum 181 am Exer 2.
2. Der 3. und der 31. Oktober entfallen feiertagsbedingt.
3. Als Ausgleich werden Feedbackgespräche für Zweiergruppen zur Abgabe der Übungen per BBB angeboten. Details nach Absprache in der Vorlesung.
4. Ab der zweiten Woche werden Übungen und Vorlesung „gleichzeitig“ stattfinden, d.h. nach Vorstellung einer Lerneinheit sind Sie gefordert, das Gelernte am Rechner auszuprobieren. Die Hoffnung ist, dass Sie alle über einen eigenen Rechner mit einer Visual Studio Installation möglichst unter MS Windows verfügen.<sup>1</sup>
5. Wenn erforderlich werde ich dabei auch durch die Reihen laufen, um Ihnen über die Schultern zu schauen.
6. Code, der in das Vorleistungspunktemodell eingehen soll, muss mir über SVN<sup>2,3</sup> zur Verfügung gestellt werden.
7. Zur Abnahme des Codes plane ich mit jeder/jedem TeilnehmerIn einen separaten Termin zur Vorstellung des Codes per BBB abzusprechen. Dauer ca. 20 Minuten.
8. Zu Beginn dürfen Sie sich in Zweiergruppen<sup>4</sup> zusammenfinden. Ab November plane ich Sie zu Teams zusammenzulegen,<sup>5</sup>
9. Die Vorlesung wendet sich zunächst an Masterstudenten. Sofern noch Plätze zur Verfügung stehen – was in den letzten Jahren immer gegeben war, und nach Rücksprache mit dem Prüfungsausschuss kann die Vorlesung auch für Bachelorstudenten mit bereits fundierten Programmierkenntnissen von Interesse sein. Die Klärung der Anrechenbarkeit mit dem Prüfungsausschuss liegt jeweils bei Ihnen.
10. Die Kopien (fast) aller verwendeten Folien werden vor Beginn der Vorlesung (in der Regel bis Mittwochabend auf meiner Homepage zur Verfügung gestellt.<sup>6,7</sup>

---

<sup>1</sup> Der Lernstoff kann aber eigentlich in jeder C++-Entwicklungsumgebung (Apple, Mac, Linux über die Kommandozeile mit g++, Eclipse mit C++-Plugin) ausprobiert werden. Meine Unterstützung bei der Einrichtung und später beim zur Verfügung stellen von fertigen Projektmappen/Makefiles ist dann aber sehr eingeschränkt.

<sup>2</sup> In den bisherigen Semestern kam SVN zum Einsatz. GIT wird aktuell (für mich) nicht vom RZ unterstützt.

<sup>3</sup> In den ersten Wochen, bis SVN stabil läuft, ist auch über E-Mail als zip-Verzeichnis möglich.

<sup>4</sup> Drei-Personen sind übrigens **keine** Zweiergruppe. Einzelabgaben sind erlaubt, machen aber natürlich meinen Aufwand größer. In Zweiergruppen muss der Anteil jeder Einzelnen / jedes Einzelnen für mich klar erkennbar sein. Das kann z.B. bei der Vorstellung des Codes dadurch erfolgen, dass unterschiedliche Codeteile von unterschiedlichen GruppenmitgliederInnen vorgestellt werden.

<sup>5</sup> Ca. 4 – 5 Teilnehmer pro Team.

Sollte das einmal nicht der Fall sein (weder SVN noch über Homepage), habe ich es wahrscheinlich vergessen und bitte Sie um eine E-Mail-Benachrichtigung ([Hartmut.Helmke@dlr.de](mailto:Hartmut.Helmke@dlr.de)).

Es handelt sich hierbei **nicht** um ein Skript, sondern die Folienkopien sollen Ihnen lediglich das Mitschreiben erleichtern. Die Folienkopien können ein Lehrbuch nicht ersetzen. Nach der Vorlesung gibt es ab und zu auch noch Aktualisierungen der Folienkopien. Auf der Homepage wird für die ersten ca. sieben Vorlesungswochen ein Skript zur Verfügung gestellt. Ggf. wird dieses für die folgenden 6 Wochen auch noch ergänzt.

11. Wir werden sofort ganz tief in C++ einsteigen. Sie werden die ersten Programmbeispiel (ohne sehr gute Vorkenntnisse) nicht im Detail verstehen, aber keine Angst, im Laufe des Semester werden wir sowohl die elementaren Sprachelementen von C++ kennenlernen, wie sie schon seit den 70er-Jahren Bestandteil von C sind, als auch, wie C++ die Programmierung mit Templates und der Standard-Template-Bibliothek (STL) unterstützt.
12. Gute Kenntnisse in Java sind hilfreich, aber nicht zwingend erforderlich. Sie wissen selbst am besten, wo noch Lücken sind. Arrays, Schleifen, Klassen sollten Sie nicht nur kennen, sondern (in Java oder einer anderen Sprache) konzeptionell **beherrschen**.
13. In der Vorlesung werde ich Ihnen (zusätzlich zu den bewerteten Übungen) Aufgaben zur eigenverantwortlichen Bearbeitung übergeben. Die Aufgaben werden nicht bewertet und sollen den Stoff der Vorlesung vertiefen, d.h. in der Klausur wird auch Stoff als bekannt vorausgesetzt, der nicht in der Vorlesung explizit behandelt wurde, sondern den Sie sich selbst bei der Lösung der Aufgaben erarbeitet haben.
14. Die bewerteten und nicht bewerteten Aufgaben können Sie in Gruppen oder allein bearbeiten.
15. Sie werden freitags Gelegenheit haben, einen Teil der Aufgaben (aber nicht alles) im Rahmen der Veranstaltung zu bearbeiten und mit mir zu diskutieren.<sup>8</sup>
16. Die Aufgaben zu den Übungen stehen jeweils rechtzeitig im Netz (siehe Homepage bzw. der Spiegel davon im SVN).
17. Die Abgabetermine sind fix. Es entscheidet das Eincheck-Datum im SVN.
18. Sie werden Ihren Code, der zum Abgabedatum im SVN eingchecked ist, auf von mir nach dem Abgabedatum zur Verfügung gestellten Testdateien selber testen und damit bewerten und mir dann Ihre erzielten Punktzahlen über das SVN mitteilen.
19. Es gelten die Regeln für Betrugsversuche.
20. Insgesamt können Sie für die Bearbeitung dieser Aufgaben bis zu 30 Punkte für die Klausur erreichen.<sup>9</sup>

---

<sup>6</sup> Durch die Umgestaltung des Internet-Auftritts der Ostfalia und damit auch des Fachbereichs ist derzeit noch nicht geklärt, wo die Seiten abgelegt werden. Im letzten Semester war es z.B.

<https://www.ostfalia.de/cms/de/pws/helmke/lehre/> bzw.

<https://www.ostfalia.de/cms/de/pws/helmke/.content/documents/WS2425.html>

<sup>7</sup> Ich werde eine Kopie der Homepage auch über das SVN spiegeln. Einstiegsseite wäre dann <https://code.ostfalia.de/svn/i-wpf-cpp/AlleGruppen/WS2526-KopieDerHomepage>. Bitte dieses Verzeichnis komplett auschecken und dann die dortige Datei WS2526.html mit einem Browser öffnen. Aktuell haben Sie darauf aber noch keinen Zugriff. Daher brauche ich Ihre Anmeldungen. Ist etwas umständlich, das Sie jeweils selber per „SVN update“ ihre lokale Kopie aktuell halten müssen. Dafür haben Sie aber auch gleiche alle relevanten Dateien auf Ihrer Festplatte.

<sup>8</sup> 52 Stunden Vorlesung reicht für 1,7 Credits. Sie streben aber fünf Credits an, 5 Credits entsprechen in etwa 150 Stunden Aufwand inklusive der Vorbereitung auf die Klausur.

21. Die Übungen bauen überwiegend aufeinander auf.
22. The exercises will mostly be formulated in a foreign language.
23. In der Klausur können Sie insgesamt 80 Punkte erhalten, sodass Sie im Extremfall durch Klausur und Übungen zusammen 110% der Punkte erreichen können. Die 1,0 ist für ca. 96% der Punkte geplant und die 4,0 ab 50%.
24. 110% gibt es als Ausgleich auch dafür, dass Sie bei einer Übung z.B. krankheitsbedingt verhindert sein könnten.
25. Die (bewerteten) Übungen werden zeitnah abschließend bewertet, wobei Ihre Vorbewertungen in den meisten Fällen zugrunde gelegt werden und die Ergebnisse werde ich Ihnen auch mitteilen.
26. Die Aufgaben und Abgabetermine werden über die Homepage (bzw. dessen Spiegel im SVN) veröffentlicht.
27. Den größten Nutzen haben Sie von der Bearbeitung der Aufgaben jedoch nicht durch die Sonderpunkte, sondern dadurch, dass Sie sich **zeitnah** mit dem Stoff vertraut machen. Die Erfahrung zeigt, dass die meisten von Ihnen die Klausur schaffen würden und zwar auch gut, wenn Sie „nur“ mehr Zeit gehabt hätten, aber die haben Sie nur vor der Klausur.

---

<sup>9</sup> Hier ist vorausgesetzt, dass eine Klausur in Präsenz möglich ist. Ansonsten werden wir eine flexible Lösung finden. 2021/22 zählten die Übungen z.B. 50% und die Klausur auch 50%.

## Terminplanung im Detail (Stand 03.08.2025)

	Termin		Vorlesung	Übungen und Feedback
Vorles	Woche		Freitag ; Block 1+2	
1	1	26. Sep	Einführung, Worthäufigkeiten in einer Datei zählen; Datentyp., Steueranw.	
	2	03. Okt		
2	3	10. Okt	Funktionen Strukturen; Test first; Header-Dateien, Include-Wächter, Namespaces, Werte- und Ref-Semantik, Zeiger, Heap- und Stackpeicher	Abgabe Fr 10.10; Erste einfache Übung 5% Pkte
3	4	17. Okt	Werte-, Zeigerparameter etc., Klassen string, vector<T>	
4	5	24. Okt		
5	6	31. Okt		Finale Abgabe Fr. 31.10 Übung 2; Demo per BBB vorher
6	7	07. Nov	tiefe und flache Kopie: Kopierkonstruktor, LogTrace ; Operatoren	7.11; Teambildung abgeschlossen
	8	14. Nov	Operatoren (Teil 2); Templates, Polymorphie	
7	9	21. Nov	STL, Iteratoren;	
8	10	28. Nov	lineare und assoziative Container; STL Algorithmen	Bis 27.11 Erste Teamaufgabe abgegeben; Demo per BBB vorher
			Algorithmus oder Methode Klasse unique_ptr,, shared_ptr, Lambda-Ausdrücke	
9	11	05. Dez		
10	12	12. Dez	Verschiebeoperatoren ,	
11	13	19. Dez	Rest; Vorbereitung Klausur	Finale Abgabe Teamaufgabe Mi. 17.12; Demo per BBB vorher

Die Inhalte der einzelnen Wochen sind unverbindlich wie das Wetter und die Lottozahlen. Bzgl. der Termine für Feedback zu den abgegeben Übungen, siehe den vorherigen Text.