

Wahlpflichtfach

„Die verschiedenen Programmierparadigmen von C++“ im WS 2024/25

Im Folgenden sind kurz die Randbedingungen der Veranstaltung „Die verschiedenen Programmierparadigmen von C++“ im WS 2024/25 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 27. September 2024 mit einem Doppelblock im ersten und zweiten Block, ab 8:15 bis 11:30 im Raum 223.
2. Der 4. Oktober und der 1. November fallen auf einen Freitag mit vorausgehendem Feiertag. Wegen mir können wir dann die Vorlesung durchführen. Wir könnten aber auch verlegen, wenn wir uns mit Mehrheit einigen können. Der Termin am 15.11. wird sehr, sehr wahrscheinlich entfallen. Um Nachholtermine anzubieten und ggf. auch vor der Weihnachtswoche aufhören zu können, werde ich zusätzlich vier bis fünf Termine zur Betreuung und/oder Abnahme von Übungsaufgaben vermutlich per BBB anbieten. Details werden wir gemeinsam in der Vorlesung besprechen. Ich bin (noch) weitgehend flexibel.
3. 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.¹
4. Ich plane, nicht durch die Reihen zu laufen, um Ihnen über die Schultern zu schauen. Hier ist daran gedacht, dass Sie mir Ihren Code über SVN/GIT² oder E-Mail zur Verfügung stellen oder dass wir über Big Blue Button (auch in der Präsenzveranstaltung) den Bildschirm teilen.
5. Die Vorlesung wendet sich zunächst an Masterstudenten. Sofern noch Plätze zur Verfügung stehen – was in den letzten Jahren immer ermöglicht wurde, 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.
6. Die Kopien (fast) aller verwendeten Folien werden vor Beginn der Vorlesung (in der Regel bis Mittwochabend auf meiner Homepage zur Verfügung gestellt:
<https://www.ostfalia.de/cms/de/pws/helmke/lehre/>
<https://www.ostfalia.de/cms/de/pws/helmke/.content/documents/WS2425.html>

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).

¹ 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.

² In den bisherigen Semestern kam SVN zum Einsatz. GIT wird aktuell nicht vom RZ unterstützt. Ich versuche hier bis Semesterbeginn eine Lösung zu finden, die meinen administrativen Aufwand in Grenzen hält.

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 acht Vorlesungswochen ein Skript zur Verfügung gestellt. Ggf. wird dieses für die folgenden 6 Wochen auch noch ergänzt.

7. 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 werden wir sehen, wie C++ die Programmierung mit Templates und der Standard-Template-Bibliothek (STL) unterstützt.
8. 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**.
9. 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.
10. Die bewerteten und nicht bewerteten Aufgaben können Sie in Gruppen oder allein bearbeiten.
11. Sie werden freitags bzw. am Alternativtermin Gelegenheit haben, einen Teil der Aufgaben (aber nicht alles) im Rahmen der Veranstaltung zu bearbeiten und mit mir zu diskutieren.³
12. Die Aufgaben zu den Übungen stehen jeweils rechtzeitig im Netz (siehe Homepage bzw. der Spiegel davon im SVN/GIT).⁴
13. Parallel gibt es bewertete Programmieraufgaben, die Sie allein (oder im Zweierteam) bearbeiten werden und über SVN mir zur Verfügung stellen.
14. Die Abgabetermine sind fix. Es entscheidet das Eincheck-Datum im SVN/GIT.
15. Sie werden Ihren Code, der zum Abgabedatum im SVN/GIT 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/GIT mitteilen.
16. Es gelten die Regeln für Betrugsversuche.
17. Insgesamt können Sie für die Bearbeitung dieser Aufgaben bis zu 30 Punkte für die Klausur erreichen.⁵
18. Die Übungen bauen überwiegend aufeinander auf.
19. The exercises will mostly be formulated in a foreign language.
20. In der Klausur können Sie insgesamt 80 Punkte erhalten, sodass Sie im Extremfall durch Klausur Übungen zusammen 110% der Punkte erreichen können. Die 1,0 ist für ca. 96% der Punkte geplant und die 4,0 ab 50%.

³ 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.

⁴ Ich werde voraussichtlich SVN verwenden, da ich für GIT keine Unterstützung von RZ erhalte.

⁵ Hier ist vorausgesetzt, dass eine Klausur in Präsenz möglich ist. Ansonsten werden wir flexible Lösungen finden. 2021/22 zählten die Übungen z.B. 50% und die Klausur auch 50%.

21. 110% gibt es als Ausgleich auch dafür, dass Sie bei einer Übung z.B. krankheitsbedingt verhindert sein könnten.
22. 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.
23. Die Aufgaben und Abgabetermine werden über die Homepage veröffentlicht.
24. 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, wenn Sie „nur“ mehr Zeit gehabt hätten, aber die haben Sie nur vor der Klausur.

Terminplanung im Detail (Stand 15.09.2024)

		Termin	Vorlesung	Übungen und Feedback
Vorles		Woche	Montag bzw. Freitag ; Block 1+2	
1	1	27. Sep Fr	Einführung, Worthäufigkeiten in einer Datei zählen; Datentyp., Steueranw.	
2	2	04. Okt Fr	Funktionen Strukturen;	
	3	07. Okt Mo	Alternativtermin für 4.10, Mo, Mi, Do Funktionen Strukturen;	Abgabe Sa 12.10; Levenshtein Distanz für zwei Wortfolgen; Unterstützung
3	3	11. Okt Fr	Header-Dateien, Include-Wächter, Namespaces, Werte- und Ref-Semantik, Zeiger, Heap- und	
4	4	18. Okt Fr	Werte- Ref-Semantik, Zeiger, Klassen string, vector<T>	
	5	21. Okt Mo		dyn C-Arrays, verschiedene Dateien, argv[] bei main; keine Bewertung, aber zusammen mit nächster Abgabe
5	5	25. Okt Fr	Werte-, Zeigerparameter etc., ADT; Klassen,	
6	6	01. Nov Fr	tiefe und flache Kopie: Kopierkonstruktor, LogTrace	
	7	04. Nov Mo	Alternativtermin für 1.11, Mo, Mi, Do, 4, 6, 7.11	Unterstützung/Vorabnahme der Übungen; Termin Fr. 8.11 tiefe, flache Kopie, minimale
7	7	08. Nov Fr	Operatoren; Templates	tiefe, flache Kopie, minimale Std-Schnittstelle; dyn Array Bis Fr 8.11
	8	15. Nov Fr	fällt sehr, sehr wahrscheinlich aus	
8	9	22. Nov Fr	STL, Iteratoren; Algorithmus versus Methode,	
	10	25. Nov Mo		Unterstützung/Vorabnahme der Übungen; Termin Di 17.12
9	10	29. Nov Fr	lineare und assoziative Container;	Callsign Extraction simple; Berechnung Raten, correction; 29.11
10	11	06. Dez Fr	Verschiebeoperatoren Klasse unique_ptr,, shared_ptr,	
	12	09. Dez Mo		Unterstützung/Vorabnahme der Übungen; Termin Di 17.12
11	12	13. Dez Fr	Polymorphie; Lambda-Ausdrücke	
	13	16. Dez Mo		finale Abgabe der Übungen; 17.12
12	13	20. Dez Fr	Vorbereitung Klausur	

Die Inhalte der einzelnen Wochen sind unverbindlich wie das Wetter und die Lottozahlen. Bzgl. der Termine insbesondere nach Feiertagen bzw. an Alternativterminen, siehe den vorherigen Text.