Objektorientierte Programmierung

(Winter 2008/2009)

Prof. Dr. Stefan Brass
Institut für Informatik

Übungsleiterin: Dipl.-Inform. Annett Thüring

Lernziele

- Programmieren können

  Selbständige Erstellung eigener Programme für gegebene Aufgaben.
- Rechner, Betriebssystem und Editor/Entwicklungsumgebung ausreichend bedienen können.
- Gegebene Programme verstehen können

  Z.B. Klausurfrage: Was gibt dieses Programm aus?
- Die syntaktische Korrektheit gegebener Programme beurteilen können (→ Syntax-Formalismen).
- Grundkonzepte von Programmiersprachen kennen, sich leicht in neue Sprachen einarbeiten können.

Zur Programmiersprache

- In dieser Vorlesung wird Programmierung anhand der Sprache "C++" gelehrt.

  Falls am Ende noch Zeit sein sollte: Kurzer Einstieg in Java.
- C++ ist
  - ◇ in der Praxis verbreitet, seit Jahren bewährt.
  - ◇ recht hardware-nah.
  - ◇ (leider) eine recht komplexe Sprache.

    Einige selten verwendete Konstrukte werden hier nicht behandelt.
- Auf die genaue Sprache kommt es nicht an.

  Sie werden noch viele andere Programmiersprachen lernen müssen.

Motivation (1)

Warum sollten Sie programmieren lernen?

- grundlegende handwerkliche Fähigkeit, im Berufsleben eines Informatikers notwendig.

  Selbst wenn Sie nicht an der Entwicklung wirklich großer Programme mitwirken, ist Programmierung z.B. auch nötig, um Inhalte von Datenbanken schön aufbereitet ins Netz zu stellen.
- komplexere Software-Werkzeuge bieten für Nicht-Standard-Aufgaben meist Möglichkeiten, die ähnlich zu einer Programmiersprache sind.

  Z.B. ist die Datenbanksprache SQL zwar keine allgemeine Programmiersprache, aber hat doch vieles gemeinsam mit Programmiersprachen. Programmierkenntnisse erleichtern das Erlernen von SQL.

Motivation (2)

Warum sollten Sie programmieren lernen? (Forts.)

- Vieles in der Informatik kann man besser verstehen, wenn man es im Prinzip auch selbst programmieren könnte.
- Es erleichtert Verhandlungen mit Programmierern, wenn man auch selber programmieren kann.
- Meine Kollegen und ich sind der Meinung, daß niemand ein Diplom/Bachelor-Abschluß in Informatik, Bioinformatik, oder Wirtschaftsinformatik bekommen sollte, der nicht programmieren kann.

Motivation (3)

Warum macht Programmieren Spaß?

- Man kann mit relativ wenig Aufwand komplexe Maschinen und virtuelle Welten (Spiele) konstruieren.
- Man kann sich das Leben im Umgang mit dem Computer etwas erleichtern.

  Manchmal gibt es stupide manuelle Tätigkeiten, die man durch ein kurzes Programm automatisieren kann.
- Den Computer beherrschen statt umgekehrt.

  Es ist nett, Computer-Experte zu sein. Dazu gehören Programmierkenntnisse, aber auch Hardware- und Betriebssystem-Kenntnisse.
- Konkrete Anwendung von Mathematik.

Motivation (4)

Warum ist Programmieren eine Herausforderung?

- Computer befolgen genau die gegebenen Regeln (das Programm), aber können sich kein bißchen selbst denken.
- Man muß bei der Programmierung also an alles denken: Alle möglichen Fälle, auch Ausnahmen.

  Ein Programm ist wie ein mathematischer Beweis, vielleicht sogar hinsichtlich der Präzision noch anspruchsvoller. Man kann kein Glied in der Kette auslassen, es gibt nichts, "was sich von selbst versteht".
- Hier muß man Perfektionist sein.
- Man muß sich in die Maschine hineindenken.

Ansprechpartner (1)

Dozent: Prof. Dr. Stefan Brass

- Email: `brass@informatik.uni-halle.de`

  Betreff-Zeile sollte mit `[oop08]` beginnen, möglichst aussagefähig.
- Büro: Von-Seckendorff-Platz 1, Raum 313
- Telefon: 0345/55-24740
- Sprechstunde: Dienstags, 12^{00}–13^{00}
- Frühere Unis: Braunschweig, Dortmund, Hannover, Hildesheim, Pittsburgh, Gießen, Clausthal.
- Oracle8 Certified Database Administrator.
- IBM Certified Advanced DBA (DB2 UDB 8.1).

Ansprechpartner (2)

Übungsleiterin: Dipl.-Inform. Annett Thüring

- Büro: Von-Seckendorff-Platz 1, Raum 319
- Sprechstunde: Mittwochs, 14^{00}–16^{00}.
- Telefon: 0345/55-24739
- Email: thuering@informatik.uni-halle.de

Sekretärin: Ramona Vahrenhold

- Büro: Von-Seckendorff-Platz 1, Raum 324
- Telefon: 0345/55-24750, Fax: 0345/55-27333
- Email: vahrenhold@informatik.uni-halle.de

Webseiten

- http://www.informatik.uni-halle.de/~brass/oop08/
  - ◇ Folien

    Die Folien werden jeweils vor der Vorlesung ins Web gestellt. Formate: (1) PDF, farbig, gross, (2) Postscript, s/w, 4:1 verkleinert.
  - ◇ Nützliche Links.
- http://studip.uni-halle.de

  .../goto.php?id=9a2dcd4287c02db073adfc92374a443d
  - ◇ Anmeldung zu Vorlesung und Übungsgruppen
  - ◇ Link zu Übungsplattform (Hausaufgaben)
  - ◇ Forum

Zeit und Ort (1)

Vorlesung (2 SWS):

- Dienstags, 10^{15}–11^{45}, Raum 3.28.

Übung am Rechner (2 SWS):

- Im PC Pool (3.32) und im Thin-Client Pool (3.34).
- Drei Übungszeiten (jeweils max. 2*25 Teilnehmer):

Nr	Tag	Zeit	Beginn
1 (3.34) / 4 (3.32)	Montag	12^{15}–13^{45}	6.10.
2 (3.34) / 5 (3.32)	Montag	14^{15}–15^{45}	6.10.
3 (3.34) / 6 (3.32)	Montag	16^{15}–17^{45}	6.10.

Falls erste Übung (Linux-Einf.) verpasst: Mittwoch, 16-18, 3.32.

Zeit und Ort (2)

Anwesenheitspflicht:

- In der Vorlesung: Keine Anwesenheitspflicht, aber es ist unklug, sie öfters zu schwänzen.

  Der Zeitaufwand für die eigene Einarbeitung in den Stoff ist eher höher. Durch Besuch der Vorlesung merkt man rechtzeitig, wenn man "abgehängt wurde", und kann dann fragen, bzw. selbst stärker nacharbeiten. Nicht aller Stoff steht auf den Folien. In der Vorlesung gibt es auch Beispiele, kleine Aufgaben, eventuell wichtige Ankündigungen, und nicht zuletzt Fragen der Hörer.
- In der Übung: Anwesenheitspflicht mit Ausnahmen.

  Sie dürfen drei Mal fehlen, nach Ermessen der Übungsleiterin auch mehr. Sie müssen in der Lage sein, abgegebene Hausaufgaben zu erklären (auch vor der ganzen Gruppe), und Fragen zu beantworten.

Zeit und Ort (3)

Anmeldung zu Übungsgruppen:

- Wir garantieren, daß Sie einen Platz in einer der Übungsgruppen bekommen.

  Wenn Sie zur Teilnahme an dieser Vorlesung verpflichtet sind. Notfalls werden zusätzliche Gruppen angeboten (zu anderen Zeiten).

- Wir können leider nicht garantieren, daß Sie einen Platz in einer bestimmten Gruppe bekommen.

- Sie müssen sich im Web über StudIP für eine Gruppe eintragen: [http://studip.uni-halle.de/].

  Direkte Links: [http://www.informatik.uni-halle.de/˜brass/oop08/].

Zeit und Ort (4)

Anmeldung zu Übungsgruppen, Forts.:

- Für die Anmeldung zu einer Übungsgruppe über StudIP brauchen Sie einen Benutzernamen und ein Passwort/Kennwort, das Ihnen mit der Immatrikulationsbescheinigung zugegangen sein müßte.

  Der Benutzername identifiziert Sie eindeutig im System (Vor- und Nachname sind nicht immer eindeutig). Das Passwort sollten nur Sie wissen. Es dient als Ausweis, daß Sie auch wirklich Sie sind.

- Bitte tragen Sie sich bei StudIP nicht nur für die Übungsgruppe Ihrer Wahl, sondern auch für die Vorlesung selbst ein.

Modulanmeldung

- Für fast alle Studiengänge ist die Modulanmeldung über die Selbstbedienungsfunktion CSS Pflicht.

  Z.B. BA Informatik 180, BA Bioinformatik, BA Wirtschaftsinformatik 180. Falls nicht über CSS, dann im Prüfungsamt.
  [http://www.verwaltung.uni-halle.de/SBStud/]
  Zusätzliche Anmeldungsliste in der Vorlesung (basierend auf StudIP).

- Die Modulanmeldung ist 14 Tage vor Vorlesungsbeginn bis 14 Tage nach Vorlesungsbeginn möglich, in diesem Semester also 22.09.2008 bis 20.10.2008.

- Die Modulanmeldung ist zwingende Voraussetzung für die spätere Anmeldung zur Prüfung.

Zeitliche Belastung

- Für dieses Modul gibt es 5 Leistungspunkte (LP).

  Pro Semester soll man 30 LP erwerben, so daß man nach 6 Semestern den Bachelor-Grad (180 LP) erworben hat. Jeder LP entspricht einer durchschnittlichen Stundenbelastung von 30 Stunden.

- Entspricht 150 Stunden studentischer Arbeitszeit:

Lernform	SWS	Stunden
Vorlesung	2	30
Praktische Übung	2	30
Hausaufgaben/direkte Nacharbeit	0	30
Selbststudium	0	45
Spezielle Prüfungsvorbereitung	0	15

Prüfungsmodalitäten (1)

- Die Prüfung erfolgt als Klausur.
- Zur Klausur wird nur zugelassen, wer die “Modulvorleistung” erfüllt, d.h. für dieses Modul
  - ◇ mindestens 50% der für Hausaufgaben erreichbaren Punkte bekommen hat, und
  - ◇ den praktischen Programmiertest bestanden hat.

    Der Programmiertest findet am 26.01.2009 in der Übung statt. Sie müssen einzeln am Rechner ein (nicht sehr komplexes) Programm in vorgegebener Zeit entwickeln, und es vorführen und erklären. Falls Sie aber vorher in den Übungen aktiv waren, und ausreichend viele der Präsenzübungen dort entsprechend erfolgreich bearbeitet haben, kann die Übungsleiterin entscheiden, daß der Programmiertest für Sie nicht mehr nötig ist.

Prüfungsmodalitäten (2)

Zur Klausur (Modulleistung):

- Bücher, eigene Notizen, Kopien sind erlaubt.

  Bücher sind nur nützlich, wenn man sie vorher gelesen hat.
- Rechner aller Art sind nicht erlaubt.

  Sie können also zu entwickelnde Programme nicht ausprobieren.
- Termin: 31.03.2009, 10^{15}-11^{45}, Raum 5.09 u.a.

  Achten Sie aber zur Sicherheit auf weitere Ankündigungen.
- Man muß sich bis vier Wochen vorher zur Klausur anmelden (nur möglich, falls zum Modul angemeldet und Modulvorleistung erfüllt).

Prüfungsmodalitäten (3)

Zur Klausur (Forts.):

- Falls Sie 65% der Klausurpunkte erreicht haben, haben Sie garantiert bestanden. Ab 95% gibt es die bestmögliche Zensur (1.0, A).

  Die Grenzen werden möglicherweise noch nach unten verschoben.
- Der Dozent behält sich das Recht vor, Extrapunkte (bis max. 5% der Klausurpunkte) zu vergeben für
  - ◇ Finden von Fehlern im Skript u.ä.
  - ◇ Außergewöhnlich gute Hausaufgabenergebnisse verbunden mit entsprechender Übungsteilnahme.

Prüfungsmodalitäten (4)

Zur Klausur (Forts.):

- Falls Sie an der Klausur teilnehmen und bestehen, gibt es keine Möglichkeit zur Zensurverbesserung.

  Das wurde durch den Senat der Universität festgelegt, es muß wohl für alle Prüfungsordnungen gelten, Ausnahmen wären nur über eine Freischussregelung möglich (nicht in Informatik/Wirtschaftsinformatik).
- Nachholtermin: (wird noch bekanntgegeben)

  Bitte informieren Sie sich über Ihre Prüfungsordnung, inwieweit eine extra Anmeldung nötig ist, oder Sie automatisch angemeldet sind, falls Sie die erste Klausur nicht bestanden haben.

Prüfungsmodalitäten (5)

Folgen des Nichtbestehens:

- Falls Sie die Nachholklausur auch nicht bestehen:
  - müssen Sie das Modul ein Jahr später neu belegen (ggf. andere Programmiersprache),
  - können Sie mehrere der für folgende Semester geplanten Lehrveranstaltungen nicht belegen.
- Es gibt maximal drei Prüfungsversuche.

  Nach dem dritten nicht erfolgreichen Prüfungsversuch für diese Vorlesung ist für Informatiker und Wirtschaftsinformatiker das Studium beendet. Im Laufe des Studiums sind nur 7 bzw. 8 zweite Wiederholungsprüfungen erlaubt, eventuell nur mit Antrag. Unentschuldigtes Fehlen bei der Klausur zählt als nicht erfolgreicher Prüfungsversuch.

Prüfungsmodalitäten (6)

Hausaufgaben (gehört zu Modulvorleistungen):

- Es gibt wöchentlich Hausaufgaben, sowohl ganz einfache Wiederholungen zum Skript als auch Programmieraufgaben.
- Programmieraufgaben müssen mit `g++` unter `Linux` lauffähig sein.

  Sonst gibt es 0 Punkte für die jeweilige Aufgabe. Sie können auch eine andere Entwicklungsumgebung nutzen (z.B. Microsoft Visual Studio), sollten dann aber die Portabilität testen. Für Abgaben ohne ausreichende Kommentare, oder mit besonders schlechtem Programmierstil (z.B. `goto`) kann es auch 0 Punkte geben.

Prüfungsmodalitäten (7)

Hausaufgaben (Forts.):

- Hausaufgaben können einzeln oder in Gruppen bis ca. 3–4 Personen bearbeitet werden.

  Gruppenarbeit muß klar gekennzeichnet sein, es ist dann auch nur ein Exemplar der Lösung abzugeben.
- Jedes Gruppenmitglied muß jede von der Gruppe abgegebene Lösung vorführen/erklären können.

  Wenn Sie die eigene Lösung nicht befriedigend erklären können, kann das als Täuschungsversuch gewertet werden!
- Damit das greift: Anwesenheitspflicht (s.o.).

  Wenn Sie mehr als drei Mal fehlen, bekommen Sie 0 Punkte für Hausaufgaben die Sie (wegen Abwesenheit) nicht erläutern konnten.

Prüfungsmodalitäten (8)

Hausaufgaben (Forts.):

- Die Hausaufgaben werden wöchentlich am Dienstag (nachmittag) ins Web gestellt und müssen bis zum Montag der folgenden Woche (24^{00}) über das StudIP-System abgegeben werden.

  Genauer: Ausgabe/Abgabe über Übungsplatform, die in das StudIP-System eingehängt ist, und von Frau Thüring mit entwickelt wurde.
- Für die theoretischen Aufgaben wird am Dienstag morgen eine Musterlösung ins Netz gestellt.
- Die praktische Aufgaben werden in der nächsten Übung besprochen (Präsentation durch Teilnehmer).

Prüfungsmodalitäten (9)

Hausaufgaben (Forts.):

- Falls Sie die Korrektur Ihrer Abgabe oder die Musterlösung nicht verstehen, fragen Sie!

  Fehler kommen leider gelegentlich vor. Selbst wenn sich am Ende herausstellt, daß der Punktabzug berechtigt war, haben bei der Diskussion beide Seiten etwas gelernt. “Wer nicht fragt, bleibt dumm.”

- Nach der Besprechung der Hausaufgaben wird in der Übung mit zusätzlichen Aufgaben geübt.

  Typischerweise Programmieraufgaben, die erst in der Übung bekannt gegeben werden, und deren Lösung innerhalb einer vorgegebenen Zeit der Übungsleiterin vorgeführt werden kann.

Prüfungsmodalitäten (10)

Hausaufgaben (Forts.):

- Bitte nicht abschreiben und nicht jemand anders zum Abschreiben geben!

  Manchmal werden komische Fehler blind abgeschrieben, davon lernt man nichts. Bei zu ähnlichen Programmen gibt es für alle Beteiligten 0 Punkte, im Wiederholungsfall kann es als Täuschungsversuch gewertet werden. Wir haben Software zur Plagiats-Erkennung!

- Erreichen Sie die geforderten Punkte nicht, müssen Sie die Vorlesung ein Jahr später erneut belegen.

  Es gibt vorher keine Wiederholungsmöglichkeit. Damit können Sie aber auch an mehreren Vorlesungen des zweiten Semesters nicht teilnehmen. Bei außergewöhnlichen Belastungen (längere Krankheit etc.) sprechen Sie mit dem Dozenten über eine mögliche Ersatzleistung.

Prüfungsmodalitäten (11)

Prüfungsergebnisse vom 16.02.2007			
1.0	sehr gut	11	17%
1.3	sehr gut (-)	6	9%
1.7	gut (+)	9	14%
2.0	gut	5	8%
2.3	gut (-)	2	3%
2.7	befriedigend (+)	5	8%
3.0	befriedigend	4	6%
3.3	befriedigend (-)	1	2%
3.7	genügend (+)	0	0%
4.0	genügend	1	2%
5.0	nicht bestanden	21	32%
5.0	nicht erschienen	8	

Prüfungsmodalitäten (12)

- Die Klausur war recht kurz nach Ende der Vorlesung, diesmal gibt es mehr Zeit zur Vorbereitung.
- Von den 21, die an der Klausur teilgenommen haben, und durchgefallen sind, haben im 2. Versuch
  - 5 bestanden (1.7, 1.7, 2.7, 3.0, 4.0),
  - 9 zum zweiten Mal nicht bestanden,
  - 7 nicht an der Nachholklausur teilgenommen.

    Möglicherweise haben sie das Modul ein Jahr später besucht und dann an der Klausur teilgenommen. Auch Studierende, die zwei Mal durchgefallen sind, hätten normalerweise noch einen Versuch.
- Eventuell gibt es ein Tutorium vor der Nachklausur.

Rechnernutzung, Software (1)

Rechnerpools:

- PC-Pool: Raum 332

  29 PCs (Athlon64 X2 4200+, 1 GB RAM, 19" TFT-Monitor).
  Windows XP Professional / Ubuntu Linux 8.04 64bit

- ThinClient-Pool: Raum 334

  Windows Server "odin" (4x Opteron 852, 16 GB, Windows 2003)
  Linux Server "anubis" (4x Opteron 852, 16 GB, Ubuntu 8.04 64bit)
  Unix Server "turing" (4x Sparc II 400 Mhz, 2 GB, Solaris 10 sparc)

- Server für Home-Verzeichnisse (odin)

  RAID-System mit 1 Terabyte, Quota pro Nutzer: 200 MB.

Rechnernutzung, Software (2)

Rechnerpools (Forts.):

- Damit Sie auf den Rechnern im Pool 3.34 arbeiten können, muß ein Benutzerkonto eingerichtet werden. Die Daten entsprechen dem StudIP-Account.

  Hierzu melden Sie sich bitte vor der ersten Übung bei der Rechneraufsicht in Raum 333 (möglichst nach 14 Uhr). Bringen Sie Ihren Studentenausweis mit. Es wird das Start-Passwort von StudIP übernommen (das Sie mit Ihrer Immatrikulation bekommen haben).

- Remote Login (per `ssh`) z.B. auf Linux-Rechner

  `anubis.informatik.uni-halle.de`

  Sie können z.B. `PuTTY` für das remote Login nutzen.

Rechnernutzung, Software (3)

Rechnerpools (Forts.):

- Raum 303: Für eigene Notebooks.
- Raum 335, 302: Uni-Pools (Anmeldung beim URZ).

Software auf Windows Rechnern:

- Microsoft Visual Studio .NET 2005
- Eclipse, GNU C++ Compiler, Cygwin

Software auf Linux Rechnern:

- Eclipse, GNU C++ Compiler.
- Sun Studio 12 (Einführung am 20.10. in Übung)

Rechnernutzung, Software (4)

- GNU C++

  [http://gcc.gnu.org]. Auf Linux-Rechnern häufig schon vorinstalliert.
  Unter Windows Cygwin Bibliothek nötig [http://www.cygwin.com/].

- Microsoft Visual C++, Express Edition (kostenlos)

  [http://www.microsoft.com/germany/msdn/vstudio/products
  /express/visualc/default.mspx]

- Borland (kostenloser Compiler, Turbo-Debugger)

  [http://www.borland.com/downloads/download_cbuilder.html]

- Eclipse (Java-basierte IDE)

  [http://www.eclipse.org/], [http://www.eclipse.org/cdt/]

- Code::Blocks (freie C++ IDE für Windows/Linux)

  [http://www.codeblocks.org/]

Lehrbücher (1)

- Ulla Kirch-Prinz, Peter Prinz:
  C++, Lernen und professionell anwenden.

  3. Aufl., mitp, 2005, ISBN 3-8266-1534-4, 895 Seiten, 44.95 Euro
  Microsoft Visual C++ Book Edition auf beiliegender CD.

- Helmut Erlenkötter: C++, Objektorientiertes Programmieren von Anfang an.

  9. Aufl., Rowohlt, 2006, ISBN 3-499-60077-3, 332 Seiten, 10.50 Euro.

- Bjarne Stroustrup:
  The C++ Programming Language.

  Special Ed., Addison-Wesley, 2000, ISBN 0-201-70073-5, 1020 Seiten, 50.89 Euro.

Lehrbücher (2)

- Dirk Louis: C++ easy.
  Programmieren mit einfachen Beispielen.

  Markt und Technik, 2005, ISBN 3827269679, 384 Seiten, mit C++ Builder X Personal Edition auf CD-ROM, 16.95.

- Scott Meyers: Effective C++. 55 Specific Ways to Improve Your Programs and Designs.

  3rd Ed., Addison-Wesley, 2005, ISBN 0-321-33487-6, 297 Seiten, 33.89 Euro.

- Scott Meyers: More Effective C++. 35 New Ways to Improve Your Programs and Designs.

  Addison-Wesley, 1996, ISBN 0-201-63371-X, 318 Seiten, 35.89 Euro.

Lehrbücher (3)

- RRZN: Programmiersprache C (Nachschlagewerk), 14. Aufl., 3.70 Euro

  Eine Auswahl der vom Regionalen Rechenzentrum Niedersachen und seiner Kooperationspartner verfassten Schriften ist im Universitätsrechenzentrum, Kurt-Mothes-Straße 1, 06120 Halle, bei Frau Oelgarte (Raum 108, Tel. 55-21815) oder bei Frau Klotz (Raum 110) erhältlich. [http://www.urz.uni-halle.de/lehrgaenge/handbuecher/]

- RRZN: C++ (Von C zu C++), 12. Aufl., 3.60

  Weitere Lehrbücher aus der Reihe sind z.B.
  Windows XP — Grundlagen, 1. Aufl., 4.65 Euro
  Internet, 5. Aufl., 5.00 Euro.
  Java 2: Grundlagen und Einführung, 4. Aufl., 6.90 Euro.

Lehrbücher (4)

- Goos, Zimmermann: Vorlesungen über Informatik 1 Grundlagen und funktionales Programmieren.

  4. Aufl., Springer, 2005, ISBN 3540244050, 400 Seiten, 26.95 Euro.

- Goos, Zimmermann: Vorlesungen über Informatik 2 Objektorientiertes Programmieren u. Algorithmen.

  4. Aufl., Springer, 2006, ISBN 3540244034, 375 Seiten, 26.95 Euro.

- Hans-Jürgen Appelrath, Jochen Ludewig: Skriptum Informatik. Eine konventionelle Einführung.

  5. Aufl., Teubner, 2000, ISBN: 3519421534, 462 Seiten, 27.90 Euro.

Lehrbücher (5)

- Brian W. Kernighan, Dennis M. Ritchie:
  The C Programming Language, 2nd Ed.
  Prentice Hall, 1988, ISBN 0-13-11-362-8, 272 Seiten, 38.45 Euro.
- Michael Seeboerger-Weichselbaum:
  Programmieren mit Eclipse 3.
  mitp, 2006, 3-8266-7379-4, 389 Seiten, mit CD-ROM, 29.95 Euro.
- INCITS/ISO/IEC 14882-2003:
  Programming Languages — C++
  Offizieller Standard. Erhältlich bei [http://webstore.ansi.org] für 30$.
- [http://haweb1.bibliothek.uni-halle.de:8080/]

Verbesserung der Lehre

- Diese Vorlesung soll kein Monolog werden:
  Fragen und Ergänzungen sind sehr willkommen.
- Gute Lehre ist für mich wichtig.
  Vorschläge zur Verbesserung der Vorlesung sind sehr willkommen.
- Korrekturen für Fehler auf den Vorlesungs-Materialien, nützliche Links für die WWW-Seite etc. werden eventuell mit Extrapunkten belohnt.
  Soweit ich das nach den Vorgaben der Universität und der Modulbeschreibung darf. In jedem Fall sind die Extrapunkte nur ein kleiner Bonus.

Ratschläge zum Studium (1)

- Professoren freuen sich über Fragen!
  Selbst wenn der Professor die Frage nicht (gleich) beantworten kann, zeugt sie von Interesse, und am Ende lernen alle etwas davon.
- Lesen Sie möglichst mehrere Bücher zum Thema der Vorlesung, nicht nur das Skript.
  Seien Sie selbständig und etwas kritisch. Man kann Dinge auch ganz ohne Vorlesung lernen. Glauben Sie nicht etwas, nur weil Sie es zufällig gehört haben. Versuchen Sie zu verstehen, nicht auswendig zu lernen.
- Nehmen Sie das Studium ernst.
  Soviel Zeit, wie Sie jetzt haben, sich fortzubilden und Bücher zu lesen, haben Sie später nie wieder. Versuchen Sie, mit 6/10 Semestern auszukommen, wenn es keine besonderen Gründe gibt.

Ratschläge zum Studium (2)

- Lernen Sie programmieren (und wirklich gut).
  Es ist das Handwerk, auf dem alles beruht.
- Lernen Sie mathematisches Denken (Formalisieren, Beweisen), und Techniken wie z.B. vollständige Induktion, Grundlagen der Algebra und Logik.
- Arbeiten Sie in Gruppen zusammen, aber sorgen Sie dafür, dass jedes Gruppenmitglied alles lernt.
- Computerspiele können süchtig machen.
- Falls Frage nach Sinn des Lebens: Ich bin Christ.