Studienordnung für den Bachelorstudiengang Informatik

Präambel

…

(1) Geltungsbereich

Diese Ordnung regelt Ziele, Inhalt und Aufbau des Bachelorstudiengangs Informatik auf Grundlage der Prüfungsordnung vom TT. Monat Jahr.

(2) Studienberatung, Studienfachberatung, und Studienverlaufsberatung

(1) Die allgemeine Studienberatung wird durch die Zentraleinrichtung Studienberatung und Psychologische Beratung durchgeführt.

(2) Die Studienfachberatung und die Studienverlausberatung werden durch die Professorinnen und Professoren des Instituts für Informatik in regelmäßigen Sprechstunden durchgeführt.

(3) Jeder bzw. jedem Studierenden ist ein persönlicher Studienberater aus dem Kreis der hauptberuflich tätigen Professoren und Professorinnen zugeordnet. Diese Zuordnung wird von der bzw. von dem Vorsitzenden des Prüfungsausschusses in geeigneter Form bekannt gemacht.

(4) Studierenden wird empfohlen, in jedem Jahr mindestens einmal die Studienberatung aufzusuchen und über den erreichten Leistungsstand sowie die Planung des weiteren Studienverlaufs zu sprechen.

(5) Jede Studentin und jeder Student soll zu Beginn des dritten Semesters zur Studienverlaufsberatung. Die Studienverlaufsberatung führt der persönliche Studienberater durch. In dieser Beratung werden unter anderem die Wahl und der Verlauf des Vertiefungsgebiets (§8) und des Nebenfachs (§9) besprochen. Datum und Inhalt der Beratung sind aktenkundig zu machen.

(6) Eine Studienfachberatung ist obligatorisch, wenn die Studentin oder der Student drei Monate nach dem Beginn des dritten Semesters noch nicht 20 Leistungspunkte erworben hat (Studienfachberatung nach BerlHG §28(3)).

(3) Ziele des Studiums

(1) Die Absolventinnen und Absolventen verfügen über solide Kenntnisse über die Grundlagen der Informatik. Sie sind mit wissenschaftlichen Arbeitsweisen und allgemeinen Methoden der Informatik vertraut. Sie sind in der Lage, mit Methoden der Informatik abstrakte Modelle zu erstellen oder solche zu konkretisieren und Modelle zu diskutieren. Sie können für vielfältige Probleme algorithmische Lösungsstrategien formulieren. Sie erkennen Zusammenhänge und Probleme im Bereich des Softwaresystementwurfs und der Softwaresystementwicklung und können für diese Lösungsvorschläge formulieren. Sie können ein Softwareprojekt geringen Umfangs selbständig planen und durchführen sowie größere Projekte anteilig im Team übernehmen, um Teilaufgaben selbständig zu bearbeiten, die Ergebnisse anderer aufzunehmen und die eigenen Ergebnisse weiterzugeben. Die Absolventinnen und Absolventen sind in der Lage, geeignete Algorithmen, Datenstrukturen und Komponenten zur Lösung von Problemen unter Berücksichtigung technischer und wirtschaftlicher Randbedingungen auszuwählen und zu kombinieren sowie die Lösung nach ihrer Qualität und Angemessenheit zu beurteilen. Sie verstehen die prinzipiellen Grenzen algorithmischer Berechenbarkeit. Die Absolventinnen und Absolventen sind sich der vielfältigen Sicherheitsprobleme bewusst, die mit dem Einsatz von informatischen Systemen einhergehen und wissen, welche Techniken und Methoden für die Sicherung von Systemen zum Einsatz kommen. Sie können weiterhin die ökonomische Bedeutung von informatischen Systemen einschätzen und sind sich gesellschaftlicher Auswirkungen bewusst. Innerhalb eines Vertiefungsgebiets sind sie mit dem Stand der Forschung, entsprechenden Methoden, Inhalten und Anwendungen vertraut. Sie können sich selbständig und

zügig in neue Anwendungsbereiche und Technologien einarbeiten. Sie können kritisch urteilen und handeln verantwortlich.

(2) Über die Qualifikationen in der Informatik hinaus besitzen die Absolventinnen und Absolventen individuelle Kenntnisse und Kompetenzen, die sie im Studium eines Nebenfachs (affiner Disziplinen) aus den Bereichen der Natur-, Geistes-, Wirtschafts-, Rechts- und Sozialwissenschaften sowie im Zuge der Allgemeinen Berufsvorbereitung erworben haben.

(3) Um die in Absatz 1 genannten Ziele zu verwirklichen, soll das Studium ein dauerhaft gültiges Grundlagenwissen in Theoretischer, Praktischer, Angewandter und Technischer Informatik vermitteln und die Studentinnen und Studenten mit wichtigen, dem Stand der Technik entsprechenden Methoden und Techniken der Informatik und ihren Anwendungen vertraut machen.

(4) Inhalte des Studiums

(1) Im Bachelorstudiengang werden auf Grundlage von mathematischen und informatischen Theorien und Methoden existierende Softwaresysteme und deren Anforderungen analysiert und formalisiert. Techniken des Entwurfs und der Verwirklichung von neuen Software- und Hardwaresysteme werden erlernt und die Qualität des Systems durch empirische, induktive und deduktive Methoden gesichert.

(2) In Algorithmen und Programmierung werden die grundlegenden Methoden zur Programmierung von Rechnern erlernt.

(3) In Technischer Informatik werden die grundlegenden Eigenschaften von Hardwaresystemen untersucht. Die Studentinnen und Studenten lernen, …

(4) In theoretischer Informatik werden die fundamentalen Möglichkeiten und Grenzen des Rechnens erlernt. Es werden anhand von Beispielen Methoden gelehrt, die eine endgültige Beurteilung der Verwirklichbarkeit eines Vorhabens oder dessen potentiellen Kosten-/Nutzen, insbesondere des Aufwands in Zeit und Speicher, zeigen.

(5) In angewandter Informatik werden Technologien und deren Verwendung gelehrt.

(6) In Mathematik für Informatik werden die grundlegenden Sprachgebräuche und Methoden des formalen Diskurs über Softwaresysteme erlernt und geübt. Die mathematischen Verfahren werden in Algorithmen überführt.

(5) Berufs- und Tätigkeitsfelder

(1) Mit dem Abschluss des Bachelorstudiums sind die Absolventinnen und Absolventen für einen weiterführenden Masterstudiengang in der Informatik qualifiziert.

(2) Das Bachelorstudium Informatik bereitet auf die berufliche Praxis auf dem Gebiet der Informatik in anwendungs-, herstellungs-, forschungs- und lehrbezogenen Tätigkeiten vor. Besondere Bedeutung kommt der Fähigkeit zu, sich auf wechselnde Aufgabengebiete einstellen zu können, sich den wandelnden Bedingungen der Praxis der Informationsverarbeitung anpassen zu können und diesen Wandel aktiv mitzugestalten.

(3) Beschäftigung finden Informatiker und Informatikerinnen vor allem bei Unternehmen der Datenverarbeitungs-/Computertechnik (Hardware und Software), bei Herstellern von Systemen der Informations- und Telekommunikationstechnik (IT-Systemen), bei Unternehmen, die Systeme und Dienstleistungen der Informations- und Telekommunikationstechnik anbieten, z.B. bei System- und Softwarehäusern, unternehmensinternen und externen Dienstleistern auf dem IT-Sektor, DV-Beratungsunternehmen sowie in informationstechnischen Abteilungen von DV-Anwenderbetrieben, also den IT-Abteilungen jeder Branche. Beschäftigungsmöglichkeiten gibt es auch im öffentlichen Dienst. Informatiker/innen bearbeiten die unterschiedlichsten Aufgabenfelder, z.B. in Forschung und Entwicklung, bei der Produktionsplanung und -steuerung, der Betriebsorganisation, der Marktforschung u.v.a.m. Daneben gewinnen die Bereiche Gesundheitswesen (E-Health), Verwaltung (E-Government) und Sicherheitstechnik an Bedeutung.

(6) Lehr- und Lernformen

Es sind folgende Lehr- und Lernformen vorgesehen:

1. Vorlesung mit Übung: Die Lehrkraft trägt den Stoff in der Vorlesung vor und erläutert ihn. Die Studentinnen und der Student vertieft den Stoff durch regelmäßige Vor- und Nachbereitung. Die Übungen finden begleitend zur Vorlesung in kleinen Gruppen statt, die nicht mehr als zwanzig Teilnehmerinnen und Teilnehmer umfassen sollen. Die Übungen werden von studentischen Tutorinnen oder Tutoren oder wissenschaftlichen Mitarbeiterinnen oder Mitarbeitern unter der

Leitung der Lehrkraft der jeweiligen Vorlesung durchgeführt. Zu einer Vorlesung erscheinen in regelmäßigen Abständen Übungsblätter mit Aufgaben, die von den Studierenden selbständig in freier Hausarbeit oder in selbstorganisierten Kleingruppen zu lösen oder zu bearbeiten sind. Die Lösungen oder Lösungsansätze werden in den Übungsgruppen vorgetragen und diskutiert. Zweck der Übungsgruppen ist sowohl die Vertiefung des Vorlesungsstoffes als auch das Erlernen und Üben von Methoden und Techniken. Ferner soll das Gespräch über Informatik, die Zusammenarbeit und die Planung der eigenen Arbeitsweise erlernt werden.

1. Praktikum: Praktika dienen dem Erwerb von Fertigkeiten, die Problemlösungsmethoden der Informatik anhand mehrerer praktischer Aufgaben erfolgreich einzusetzen. Das schließt die Problemspezifikation und die Zerlegung in Teilprobleme ein. Lösungsvorschläge und Ergebnisse sind regelmäßig vorzubereiten, vorzuführen, schriftlich auszuarbeiten und vorzutragen. Zweck der Praktika ist der sichere Umgang mit dem erlernten Wissen und den geübten Fertigkeiten.
2. Proseminar: In einem Proseminar wird ein spezielles Thema der Informatik oder der Anwendungen der Informatik von den Studierenden und der Lehrkraft gemeinsam erarbeitet. Dazu bereitet jede Studentin und jeder Student unter Anleitung der Lehrkraft ein Referat vor, das schriftlich ausgearbeitet und im Proseminar vorgetragen und anschließend diskutiert wird. Da jedes Referat mit anschließender Diskussion mindestens etwa 45 Minuten in Anspruch nimmt, sollen Proseminare fünfzehn bis maximal dreißig Studierende umfassen. Zweck eines Proseminars ist das Erlernen gründlicher wissenschaftlicher Arbeit unter Anleitung, das Schreiben einer wissenschaftlichen Arbeit in Vorbereitung auf die Bachelorarbeit sowie der Erwerb kommunikativer Kompetenzen und rhetorischer Fertigkeiten.
3. Projekt: Projekte sind Lehrveranstaltungen, in denen ein größeres, meist anwendungsorientiertes Problem theoretisch und praktisch in einer Weise gelöst werden soll, die einer realen Situation soweit wie möglich entspricht. Das schließt Problemspezifikation, Zerlegung in Teilprobleme, Festlegung von Schnittstellen sowie Einsatz von Projektmanagementmethoden ein. Neben dem Erwerb von Fertigkeiten zur selbständigen Anwendung von Problemlösungsmethoden der Informatik auf eine konkrete Aufgabe dient ein Projekt auch der Vertiefung von kooperativen Arbeitstechniken. Gut dokumentierte, lauffähige Programme und ein zusammenfassender Projektbericht, aus dem die eigenen Leistungen hervorgehen, sind zum Abschluss des Projekts vorzulegen.
4. Seminar: In einem Seminar wird ein spezielles Themenfeld von den Teilnehmerinnen und Teilnehmern und der Dozentin oder dem Dozenten gemeinsam erarbeitet. Dazu bereitet jede Studentin und jeder Student weitgehend selbständig ein Referat vor, das schriftlich ausgearbeitet und im Seminar vorgetragen und diskutiert wird. Da jedes Referat meist eine Stunde in Anspruch nimmt, sollen Seminare fünfzehn bis maximal zwanzig Teilnehmerinnen und Teilnehmer umfassen. Zweck eines Seminars ist das Erlernen selbständiger wissenschaftlicher Arbeit sowie die Weiterentwicklung kommunikativer Kompetenzen und rhetorischer Fertigkeiten.

(7) Aufbau und Gliederung des Studiengangs

(1) Der Bachelorstudiengang Informatik gliedert sich in
1. Das Kernfach (§8)
2. Das Vertiefungsgebiet (§9)
3. Das Nebenfach (§10)
4. Die allgemeine Berufsvorbereitung (§11)

(2) Die Summe der Leistungspunkte im Vertiefungsgebiet und des Nebenfachs darf 30 LP nicht überschreiten.

(3) Über Inhalte und Qualifikationsziele, Lehr- und Lernformen, den zeitlichen Arbeitsaufwand, die Formen der aktiven Teilnahme, die Regeldauer und die Angebotshäufigkeit informieren die Modulbeschreibungen gemäß Anlage 1.

(4) Über den empfohlenen Verlauf des Studiums unterrichten die exemplarischen Studienverlaufspläne gemäß Anlage 2.

(8) Kernfach

(1) Das Kernfach des Bachelorstudiengangs umfasst 120 LP und besteht aus:
1. Studienbereich Algorithmen und Programmierung im Umfang von 32 LP mit folgenden Modulen:
   - Funktionale und Prozedurale Programmierung (8 LP)
   - Objektorientierte Programmierung (8 LP)
   - Algorithmen und Datenstrukturen (8 LP)
   - Nichtsequentielle und Netzprogrammierung (8 LP)

2. Studienbereich Technische Informatik im Umfang von 20 LP mit folgenden Modulen:
   - Grundlagen der technischen Informatik (5 LP)
   - Rechnerorganisation und Rechnerarchitektur (5 LP)
   - Betriebssysteme und Netzwerke (5 LP)
   - Hardwarepraktikum (5 LP)
3. Studienbereich Praktische Informatik im Umfang von ~~19~~ 20 LP mit folgenden Modulen:
   - Datenbanksysteme (7 LP)
   - Softwaretechnik (~~7~~8 LP)
   - Anwendungssysteme (Auswirkungen der Informatik) (5 LP)
4. Studienbereich Theoretische Informatik im Umfang von 7 LP mit folgenden Modulen:
   - Grundlagen der Theoretischen Informatik (7 LP)
5. Studienbereich Mathematik für Informatiker im Umfang von 24 LP, 28 LP wenn das Nebenfach Mathematik gewählt wurde (siehe auch §10 (2)), mit folgenden Modulen:
   - Logik und Diskrete Mathematik (8 LP)
   - Analysis (8 LP) oder Analysis I (10 LP)
   - Lineare Algebra (8 LP) oder Lineare Algebra I (10 LP)
6. Ein Proseminar Informatik (5 LP)
7. Der Bachelorarbeit (1~~5~~2 LP)

(9) Vertiefungsgebiet

Im Rahmen des Vertiefungsgebiets sind Module im Umfang zwischen ~~10~~ 8 und ~~20~~ 17 Leistungspunkten zu absolvieren. Dafür kommen nur Module in Betracht, die in Studien- und Prüfungsordnungen geregelt sind. Die Module des Vertiefungsgebiets können dem Angebot des Masterstudiengangs Informatik entnommen werden. Mindestens eines der Module muss einem der Bereiche Praktische Informatik, Technische Informatik oder Theoretische Informatik zugehören und eine Vorlesung sowie eine Übung als Lehr- und Lernformen aufweisen. Für diese Module wird auf die Studienordnung und die Prüfungsordnung für den Masterstudiengang Informatik verwiesen.

(10) Nebenfach

(1) Im Rahmen eines Nebenfachs sind Module im Umfang zwischen 13 und 22 Leistungspunkten zu absolvieren. Als Nebenfach kommt grundsätzlich jedes wissenschaftliche Studienfach in Betracht. Die Module des Nebenfachs müssen aus dem Bachelorstudiengangs eines anderen Studienfachs stammen.

(2) Studierende des Nebenfachs Mathematik sollen statt „Analysis“ das Modul „Analysis 1“ und statt „Lineare Algebra“ das Modul „Lineare Mathematik 1“ absolvieren. Die Module „Diskrete Mathematik 1“, „Analysis 1“ und „Lineare Algebra 1“ können nicht als Module eines Nebenfachs absolviert werden, sondern nur als Module des Kernfachs im Bereich „Informatiknahe Mathematik“ (§9 (1), Punkt 5).

(3) Studierende des Nebenfachs Bioinformatik absolvieren das Modul „Algorithmische Bioinformatik“.

(4) Für Module anderer Nebenfächer wird auf die Studienordnung und die Prüfungsordnung für den Bachelorstudiengang des Nebenfachs in der jeweiligen Fassung verwiesen.

(11) Allgemeine Berufsvorbereitung

(1) Im Studienbereich Allgemeine Berufsvorbereitung (ABV) werden über die fachwissenschaftlichen Studien hinaus überfachliche Schlüsselkompetenzen oder weitere für die berufliche Tätigkeit und wissenschaftliche Qualifikation nützliche Kenntnisse und Fertigkeiten erworben.

(2) Im Studienbereich ABV sind Module im Umfang von mindestens 30 LP zu belegen; davon müssen mindestens 10 LP im Berufspraktikum (§ 12) erworben werden.

(3) Im Kompetenzbereich „Fachnahe Zusatzqualifikationen“ muss das Modul Softwarepraktikum (10 LP) absolviert werden.

(4) Die Module des Studienbereichs ABV und darin erbrachte Leistungen dürfen nicht mit Modulen und Leistungen des Kernfaches gemäß § 8, dem Vertiefungsgebiet gemäß § 9 und den gewählten Modulangeboten aus dem Nebenfach gemäß § 10 übereinstimmen.

(5) Den Studentinnen und Studenten wird rechtzeitig und in geeigneter Form bekannt gegeben, welche weiteren Module des Studienbereichs ABV sie im Rahmen des Bachelorstudiengangs absolvieren

können. Es werden besonders Module zu Betriebswirtschaftlichen Grundlagen und zu Rechtlichen Aspekten der Informatik empfohlen.

(6) Für alle Module im Studienbereich "Allgemeine Berufsvorbereitung" außer für diejenigen gemäß Absatz 3 wird auf die Studienordnung und die Prüfungsordnung für den Studienbereich Allgemeine Berufsvorbereitung in Bachelorstudiengängen der Freien Universität Berlin (StO-ABV und PO-ABV) verwiesen.

(12) Berufspraktikum

(1) Das im Rahmen des Studienbereichs ABV zu absolvierende Berufspraktikum soll den Studentinnen und Studenten einen Einblick in mögliche Berufs- und Tätigkeitsfelder eröffnen und sie mit den Anforderungen der Praxis konfrontieren. Es dient der Überprüfung der erworbenen Kenntnisse und hat damit eine Orientierungsfunktion für eine zielorientierte und berufsqualifizierende Ausrichtung des Studiums.

(2) Bei der Suche nach einem geeigneten Praktikumsplatz ist die Eigeninitiative der Studentinnen und Studenten gefordert. Die Dozentinnen und Dozenten des Instituts für Informatik bemühen sich in Zusammenarbeit mit den Studentinnen und Studenten um die Erschließung geeigneter Praktikumsplätze.

(3) Der Fachbereichsrat bestellt die Praktikumsbeauftragte oder den Praktikumsbeauftragten.

(4) Beginn und Ende eines Praktikums sind der oder dem Praktikumsbeauftragten anzuzeigen.

(5) Über das Praktikum ist ein Praktikumsbericht zu schreiben. Der Bericht muss spätestens drei Monate nach dem Ende des Praktikums bei der oder dem Praktikumsbeauftragten abgegeben werden.

(13) Auslandsstudium

(1) Die Freie Universität Berlin ermuntert Studentinnen und Studenten, einen Teil ihres Studiums im Ausland durchzuführen. Das Auslandsstudium stellt eine Bereicherung der Erfahrungen dar, sowohl im akademischen Bereich, da man den Studienbetrieb an einer fremden Hochschule kennen lernt, als auch im kulturellen und persönlichen Bereich.

(14) Um eine Verzögerung des Studienfortschritts durch ein Auslandssemester zu vermeiden bzw. so gering wie möglich zu halten, werden die im Ausland erbrachten Studien- und Prüfungsleistungen auf der Grundlage eines individuellen Lernvertrags (learning agreement) zwischen den beiden Hochschulen und dem Studierenden anerkannt.

(15) Im Prinzip kann man ein Auslandsstudium auf eigene Initiative an jeder beliebigen Hochschule im Ausland absolvieren. Es gibt daneben ein Direktaustauschprogramm mit Partneruniversitäten in der ganzen Welt. Im Fachbereich Mathematik und Informatik berät die bzw. der Beauftragte für Auslandsstudien alle Studierenden, die sich für ein Auslandsstudium interessieren.

(16) Im Bachelorstudiengang Informatik empfiehlt sich besonders das 5. Fachsemester für ein Auslandsstudium, weil Wahlmodule im Vertiefungsgebiet oder im Nebenfach flexibel angerechnet werden können. Daneben gibt es auch die Möglichkeit, das Berufspraktikum im Rahmen eines Auslandsaufenthaltes zu absolvieren. Dazu berät die bzw. der Praktikumsbeauftragte.

(17) Inkrafttreten

…

Anlage 1: Modulbeschreibungen

Erläuterungen:

Die folgenden Modulbeschreibungen benennen für jedes Modul des Bachelorstudiengangs Informatik

- die Bezeichnung des Moduls
- Inhalte und Qualifikationsziele des Moduls
- Lehr- und Lernformen des Moduls
- den studentischen Arbeitsaufwand, der für die erfolgreiche Absolvierung eines Moduls veranschlagt wird
- Formen der aktiven Teilnahme
- die Regeldauer des Moduls

Die Angaben zum zeitlichen Arbeitsaufwand berücksichtigen insbesondere

- die Teilnahme im Rahmen der Präsenzstudienzeit
- die Zeit für eine eigenständige Vor- und Nachbereitung
- den Arbeitszeitaufwand für die Bearbeitung von Übungsaufgaben
- die unmittelbare Vorbereitungszeit für die Prüfung

Die Zeitangaben zum Selbststudium (unter anderem Vor- und Nachbereitung, Prüfungsvorbereitung) stellen Richtwerte dar und sollen den Studentinnen und Studenten Hilfestellung für die zeitliche Organisation ihres modulbezogenen Arbeitsaufwands bieten.
Die Angaben zum Arbeitsaufwand korrespondieren mit der Anzahl der dem jeweiligen Modul zugeordneten Leistungspunkte als Maßeinheit für den studentischen Arbeitsaufwand, der für die erfolgreiche Absolvierung des Moduls in etwa zu erbringen ist.
Die aktive Teilnahme ist neben der regelmäßigen Teilnahme an den Veranstaltungen (soweit gefordert) und der erfolgreichen Absolvierung der Prüfungsleistungen eines Moduls Voraussetzung für den Erwerb der dem jeweiligen Modul zugeordneten Leistungspunkte.
Die Anzahl der Leistungspunkte sowie weitere prüfungsbezogene Informationen zu jedem Modul sind der Anlage 1 der Prüfungsordnung für den Bachelorstudiengang Informatik zu entnehmen.

Modul: Funktionale Programmierung	[1]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Keine	

Qualifikationsziele:
Die Studentinnen und Studenten kennen am Ende des Moduls:
Die Studentinnen und Studenten können am Ende des Moduls: Beschreibungen und Quelltexte elementarer Algorithmen lesen und verstehen, elementare Algorithmen funktional und prozedural entwerfen, Anforderungen an funktionale und prozedurale Programme spezifizieren, elementare Algorithmen an neue Anforderungen anpassen, gut strukturierte Programme entwickeln, Eigenschaften von Programmen formal beweisen. Sie haben ein grundlegendes Verständnis der Berechenbarkeit.

Inhalte:
Studentinnen und Studenten erlernen die Grundlagen des Programmieren im Kleinen anhand funktionaler und prozeduraler Programmiersprachen. Es werden die Grundlagen der Berechenbarkeit (Lambda-Kalkül, primitive Rekursion, Fixpunkte), eine Einführung in die Programmiersprachen (Syntax (Backus-Naur-Form), operationale Semantik, Daten und Programm) gegeben. Danach werden Konzepte funktionaler Programmierung (primitive Datentypen, Listen, Tupel, Zeichenketten, Ausdrücke, Funktionsdefinition, Rekursion, Funktionsabstraktion, Closure, Funktionen höherer Ordnung, universelle Polymorphie) und deren Ausführung (Auswertungsstrategien) eingeführt. Es werden Techniken zum Beweisen von Programmeigenschaften (Termersetzung, strukturelle Induktion, Terminierung, Church-Rosser-Theorem) und deren Anwendungen (Typsysteme, Typherleitung und Typüberprüfung) eingeführt.Grundlegende Abstraktionen wie algebraische und abstrakte Datentypen und modularer Programmentwurf, sowie die Integration von Nebenwirkungen in funktionale Sprachen (z.B. durch Monaden) anhand von Ein- und Ausgabe runden den ersten Teil ab.

Der zweite Teil des Moduls führt in das imperative, prozedurale Programmieren und deren Daten ein. Von den Grundlagen der Berechenbarkeit (universelle Registermaschinen, Syntax und operationelle Semantik imperativer Programmiersprachen) wird der strukturierte Programmentwurf durch schrittweise korrekte Programmentwicklung anhand formaler Spezifikationen mit Vor- und Nachbedingungen sowie Invarianten eingeführt. Der Hoare-Kalkül (partielle und totale Korrektheit) oder der *weakest precondition calculus* dienen als Beweismethode. Es werden Prozeduren und Formen der Parameterübergabe (Wertübergabe, Referenzübergabe, Namensübergabe) sowie Probleme mit Prozeduren höherer Ordnung vorgestellt.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)
Vorlesung	4	Keine	Präsenzzeit Vorlesung 60 Präsenzzeit

1 Hier bitte frei lassen.

			Übung Selbststudium Prüfungsvorbereitung	30 120 30
Übung	2	Schriftliche Bearbeitung von Übungsblätter Zwei mündliche Präsentationen von Übungsaufgaben		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	An den Übungen und dem Labor			
Arbeitszeitaufwand insgesamt	240 Stunden	8 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Wintersemester			
Verwendbarkeit				

Modul: Objektorientierte Programmierung	[2]

Hochschule/Fachbereich/Institut: Freie Universität Berlin/Mathematik und Informatik/Informatik

Modulverantwortliche/er:

Zugangsvoraussetzungen: Keine

Qualifikationsziele:

Die Studentinnen und Studenten kennen am Ende des Moduls: den Unterschied zwischen Untertypen und Vererbung; den Unterschied zwischen parametrischem Polymorphismus, Überladen und später Bindung; Trade-offs der Softwareentwicklung und deren Notwendigkeit.
Die Studentinnen und Studenten können am Ende des Moduls: Externe Faktoren (z.B. Benutzer, Umwelt, Plattform) von Software verstehen, geeignet beschreiben und modellieren; Programme und Systeme objektorientiert entwerfen und geeignete Schnittstellen spezifizieren; Beschriebene Programmteile (Funktionen, Prozeduren, Klassen, Komponenten) geeignet in neuen Systemen wiederverwenden oder aus diesen neue Systeme erzeugen; Entwurfsmuster als Abstraktion richtig einsetzen; Software so entwickeln, dass sie leicht auf andere Plattformen portiert werden können; Software für verschiedene Zielgruppen dokumentieren

Inhalte:

Im Gegensatz zu *Algorithmen und Programmierung I* führt dieses Modul in die objektorientierte Programmierung und Programmierung im Großen ein. Ausgangspunkt ist das Geheimnisprinzip und seine Bedeutung für die Strukturierung von Programmen und die Konstruktion von Datenobjekten mittels Klassen. Eine zentrale Rolle bei der Modellierung von Daten spielt der Begriff der Datenabstraktion, verbunden mit der Unterscheidung zwischen Spezifikation und Implementierung abstrakter Datenobjekte und Datentypen.

- Methoden (Prozeduren und Funktionen), Parameterübergabe, Überladung
- Module, Klassen, Objekte
- Klassenhierarchien, Vererbung, abstrakte Klassen, Schnittstellen
- Entwurfsmuster (Iteratoren, Observer, MVC, ...)

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	4	-	Präsenzzeit in der Vorlesung	60
			Präsenzzeit Übung	30
			Selbststudium	120
			Prüfungsvorbereitungszeit	30

2 Hier bitte frei lassen.

Übung	2	schriftliche Bearbeitung der Übungsblätter zwei mündliche Präsentationen der Lösung jeweils einer Übungsaufgabe in der Übung
Veranstaltungssprache	Deutsch	
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung	
Arbeitszeitaufwand insgesamt	240 Stunden	8 LP
Dauer des Moduls	1 Semester	
Häufigkeit des Angebots	Jedes Sommersemester	
Verwendbarkeit		

Modul: Algorithmen und Datenstrukturen	[3]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/Mathematik und Informatik/Informatik	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Imperative und Objektorientierte Programmierung	

Qualifikationsziele:

Die Studentinnen und Studenten beherrschen den Umgang mit Datenabstraktion, Vererbung und polymorphen Typsystemen und sind in der Lage,

- abstrakte Datentypen zu spezifizieren und zu implementieren,

- Korrektheitsbeweise für die Implementierungen abstrakter Datentypen durchzuführen,

- unter Einbeziehung von Effizienzanalysen eine Entscheidung über die jeweils zu wählende Datenrepräsentation zu treffen.

Sie kennen die wichtigsten abstrakten Datentypen und ihre gängigen Implementierungen sowie die entsprechenden Schnittstellen aus den Bibliotheken der verwendeten Programmiersprache.

Inhalte:

Folgen, Mengen, Relationen, Bäume, Graphen und geometrische Objekte werden als abstrakte Typen eingeführt. Anschließend werden effizient manipulierbare Repräsentationen dieser Typen betrachtet und die zugehörigen Algorithmen auf ihre Komplexität hin untersucht.

Technische Aspekte der Datenspeicherung im Arbeitsspeicher (Keller und Halde) und im Hintergrundspeicher (Dateien, persistente Objekte) werden behandelt. Programmiert wird sowohl in objektorientierten als auch in funktionalen Sprachen.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	4	-	Präsenzzeit Vorlesung Präsenzzeint Übung Selbststudium Prüfunsvorbereitung	60 30 120 30
Übung	2	schriftliche Bearbeitung der Übungsblätter zwei mündliche Präsentationen der Lösung jeweils einer Übungsaufgabe in der Übung		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur an den Übungen			
Arbeitszeitaufwand	240 Stunden	8 LP		

3 Hier bitte frei lassen.

insgesamt		
Dauer des Moduls	1 Semester	
Häufigkeit des Angebots	Jedes Wintersemester	
Verwendbarkeit		

Modul: Nichtsequentielle Programmierung	[4]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Imperative und Objektorientierte Programmierung und Rechnerarchitektur	

Qualifikationsziele:

Die Studentin oder der Student kann am Ende des Moduls:

- nichtsequentielle Programme mit Prozessen/Threads/aktive Objekte geeignet strukturieren
- durch geeignete Synchronisationsverfahren unerwünschte nichtdeterministische Effekte sowie Verklemmungen vermeiden
- relevante Interaktionsparadigmen wie Client/Server und Peer-to-Peer unterscheiden und eigene Anwendungen nach diesen Paradigmen geeignet einordnen.
- verteilte Systeme auf der Basis von Interprozesskommunikation, Fernaufrufen und Sockets konstruieren
- Webanwendungen, Kunde/Dienstleister-Anwendungen, Peer to Peer Anwendungen geeinget entwerfen, strukturieren und realisieren.
- verteilte Systeme mit Hilfe geeigneter Middleware entwickeln
- Eigenschaften von Prozessen und Threads formal spezifizieren und diese verifizieren.

Inhalte:

Programmieren und synchronisieren von gleichzeitig laufenden Prozessen, die auf gemeinsamen Speicher zugreifen oder über Nachrichtenaustausch interagieren.

- Nichtsequentielle Programme und Prozesse in ihren verschiedenen Ausprägungen, Nichtdeterminismus, Determinierung
- Synchronisationsmechanismen: Sperren, Monitore, Wachen, Ereignisse, Semaphore
- Nebenläufigkeit und Objektorientierung
- Ablaufsteuerung, Auswahlstrategien, Prioritäten, Umgang mit Verklemmung
- Koroutinen, Implementierung, Mehrprozessorsysteme
- Interaktion über Nachrichten
- Middleware, Fernaufrufe
- Client-Server, Peer-to-Peer
- Webanwendungen
- Agentensysteme, Koordinierungsmechanismen wie Linda

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	4	-	Präsenzzeit Vorlesung Präsenzzeit Übung Selbststudium Prüfungsvorbereitung	60 30 120 30
Übung	2	Schriftliche Bearbeitung der Übungsblätter mündliche Präsentation der Lösungen von Übungsaufgaben in den Übungen		
Veranstaltungs	Deutsch			

4 Hier bitte frei lassen.

sprache		
Pflicht zur regelmäßigen Teilnahme	Nur an den Übungen	
Arbeitszeitaufwand insgesamt	240 Stunden	8 LP
Dauer des Moduls	1 Semester	
Häufigkeit des Angebots	Jedes Sommersemester	
Verwendbarkeit		

Modul: Grundlagen der technischen Informatik	[5]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Rechnerarchitektur	

Qualifikationsziele:

Die Studentinnen und Studenten können am Ende des Moduls:

- logische Funktionen auf physikalische Schaltkreise abzubilden,
- einfache Schaltungen zu verstehen und zu berechnen,
- den Einsatz der Halbleitertechnik in Schaltungen nachzuvollziehen und
- den Übergang von der analogen zur digitalen Welt und umgekehrt zu beschreiben.

Inhalte:

Das Modul Grundlagen der Technischen Informatik bildet die Basis für das Verständnis der Funktionsweise realer Rechnersysteme. Ausgehend von der Logik werden in diesem Modul vorrangig die Themenbereiche Schaltnetze und Schaltwerke, Logikminimierung, Gatter, Flip-Flops, Speicher, Automaten und einfacher Hardware-Entwurf behandelt. Weiterhin werden grundlegende Kenntnisse aus den Bereichen Halbleiter, Transistoren, CMOS, Operationsverstärker, A/D- und D/A-Umsetzer vermittelt, soweit sie für die Informatik notwendig sind.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	2	-	Präsenszeit Präsenzzeit Übung Selbststudium Prüfungsvorbereitung	30 30 60 30
Übung	2	- schriftliche Bearbeitung der Übungsblätter - zwei mündliche Präsentationen der Lösung jeweils einer Übungsaufgabe in der Übung		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur an den Übungen			
Arbeitszeitaufwand insgesamt	150 Stunden	5 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Wintersemester			
Verwendbarkeit				

5 Hier bitte frei lassen.

Modul: Rechnerarchitektur	[6]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Keine	

Qualifikationsziele:

Die Studentinnen und Studenten kennen am Ende des Moduls:

- kennen die grundlegenden Architekturmerkmale von Rechnersysteme
- die elementaren Möglichkeiten der Beschleunigung von Rechnersystemen

Die Studentinnen und Studenten können am Ende des Moduls:

- Rechner auf Assembler-Ebene programmieren
- Vor- und Nachteile der verschiedenen Mechanismen beurteilen
- die Interaktionen der Architekturmerkmale in Mehrkern- und Mehrprozessorsysteme verstehen

Inhalte:

Themenbereiche sind hier insbesondere Harvard/v. Neumann-Architektur, Mikroarchitektur RISC/CISC, Mikroprogrammierung, Pipelining, Cache, Speicherhierarchie, Bussysteme, Assemblerprogrammierung, Multiprozessorsysteme, VLIW, Sprungvorhersage. Ebenso werden interne Zahlendarstellungen, Rechnerarithmetik und die Repräsentation weiterer Datentypen im Rechner behandelt.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	2	-	Präsenzzeit Vorlesung Präsenzzeit Übung Selbststudium Prüfungsvorbereitung	30 30 60 30
Übung	2	Schriftliche Bearbeitung der Übungsblätter mündliche Präsentation der Lösungen von Übungsaufgaben in den Übungen		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur an den Übungen			
Arbeitszeitaufwand insgesamt	150 Stunden	5 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Wintersemester			
Verwendbarkeit				

6 Hier bitte frei lassen.

Modul: Betriebssysteme und Kommunikationssysteme	[7]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Rechnerarchitektur	

Qualifikationsziele:

Die Studentin oder der Student kennt am Ende des Moduls:

- die Grundlagen aktueller Betriebssysteme
- die Funktion und den Aufbau des Internets

Die Studentin oder der Student kann am Ende des Moduls:

- die Verbindung zwischen Rechnerhardware und Anwendungssoftware verstehen
- kennt die wesentlichen internen Mechanismen von Betriebssystemen
- Grundlagen von Kommunikationssystemen verstehen

Inhalte:

Themen sind daher Ein-/Ausgabe-Systeme, DMA/PIO, Unterbrechungsbehandlung, Puffer, Prozesse/Threads, virtueller Speicher, UNIX und Windows, Shells, Utilities, Peripherie und Vernetzung, Netze, Medien, Medienzugriff, Protokolle, Referenzmodelle, TCP/IP, grundlegender Aufbau des Internets.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	2	-	Präsenzzeit Vorlesung Präsenzzeit Übung Vorbereitung Prüfung Selbststudium	30 30 30 60
Übung	2	Schriftliche Bearbeitung der Übungsblätter mündliche Präsentation der Lösungen von Übungsaufgaben in den Übungen		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung			
Arbeitszeitaufwand insgesamt	150 Stunden	5 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Sommersemester			
Verwendbarkeit				

7 Hier bitte frei lassen.

Modul: Hardwarepraktikum	[8]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Rechnerarchitektur	

Qualifikationsziele:

~~Die Studentinnen und Studenten kennen am Ende des Moduls:~~

Die Studentinnen und Studenten können am Ende des Moduls:

- Analoge Schaltungen aufbauen und analysieren
- In Assembler und C hardwarenah programmieren
- Eingebettete Systeme in Betrieb nehmen
- Software auf eingebetteten Systemen installieren.

Inhalte:

Das Modul Hardwarepraktikum vertieft mit zahlreichen praktischen Übungen das in den Modulen Rechnerarchitektur und Betriebs- und Kommunikationssysteme Erlernte. Aufbauend auf einer einfachen Hardwareplattform mit Prozessor und diversen Schnittstellen sollen die Teilnehmerinnen und Teilnehmer lernen, elementare Treiber zu programmieren, Betriebssystemroutinen zu erweitern und die Schnittstellen anzusteuern. Anschließend sollen die Systeme vernetzt werden und mit ihrer Umwelt in Interaktion treten können.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Praktikum	2	Erfolgreiches absolvieren von 85% der Versuche	Präsenszeit Vor- und Nachbereitung Versuchsberichte	30 90 30

Veranstaltungssprache	Deutsch	
Pflicht zur regelmäßigen Teilnahme	Ja	
Arbeitszeitaufwand insgesamt	150 Stunden	5 LP
Dauer des Moduls	1 Semester	
Häufigkeit des Angebots	Jedes Wintersemester	
Verwendbarkeit		

8 Hier bitte frei lassen.

Modul: Grundlagen der Theoretischen Informatik	[9]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Keine	

Qualifikationsziele:

Die Studentinnen und Studenten kennen am Ende des Moduls:

- die theoretischen Grundlagen zur Beschreibung und syntaktischen Analyse von Programmiersprachen
- die Gleichartigkeit unterschiedlicher Beschreibungsformen von Berechenmodellen

Die Studentinnen und Studenten können am Ende des Moduls:

- die prinzipiellen Möglichkeiten und Grenzen der Berechenbarkeit verstehen.

Inhalte:

Theoretische Rechenmodelle: Automaten, Turing-Maschinen. Formale Sprachen, Sprachakzeptoren, reguläre Ausdrücke, Grammatiken, Chomsky-Hierarchie. Turing-Maschinen, Berechenbarkeit; Komplexität.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	3	-	Präsenzzeit Vorlesung Präsenzzeit Übung Vor- und Nachbereitungszeit Vorbereitung zur Prüfung	45 30 105 30
Übung	2	Schriftliche Bearbeitung der Übungsblätter mündliche Präsentation der Lösungen von Übungsaufgaben in den Übungen		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung			
Arbeitszeitaufwand insgesamt	210 Stunden	7 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Sommersemester			
Verwendbarkeit				

9 Hier bitte frei lassen.

Modul: Datenbanksysteme	[10]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Keine	
Qualifikationsziele: Die Studentinnen und Studenten können am Ende des Moduls: Datenbanken nach dem Stand der Kunst zu entwerfen und sie mit Hilfe eines Datenbanksystems zu implementieren und Anwendungen, die Datenbanken nutzen, realisieren; interne Abläufe eines Datenbanksystems und dessen Architektur verstehen.	
Inhalte: Datenbankentwurf mit Entity-Releationship Modellen und der UML; theoretische Grundlagen relationaler Datenbanksysteme, relationale Algebra; funktionale Abhängigkeiten, Normalformen, relationale Datenbankentwicklung: Datendefinition, Fremdschlüssel, andere Integritätsbedingungen, objektrelationale Abbildung, Sicherheits- und Schutzkonzepte; Transaktionsbegriff, transaktionale Garantien, Synchronisierung des Mehrbenutzerbetriebs, Fehlertoleranzeigenschaften.	

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	3	-	Präsenzzeit Vorlesung Präsenzzeit Übung Vor- und Nachbereitungszeit Vorbereitung zur Prüfung	45 30 105 30
Übung	2	Schriftliche Bearbeitung der Übungsblätter mündliche Präsentation der Lösungen von Übungsaufgaben in den Übungen		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung			
Arbeitszeitaufwand insgesamt	210 Stunden	7 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Sommersemester			
Verwendbarkeit				

10 Hier bitte frei lassen.

Modul: Proseminar Informatik	[11]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Keine	

Qualifikationsziele:

Die Studentinnen und Studenten können am Ende des Moduls:

- sich unter Anleitung in ein Thema der Informatik anhand von wissenschaftlicher Literatur einarbeiten
- beherrschen gängige Vortrags und Präsentationstechniken
- können einen Vortrag schriftlich Ausarbeiten
- können eine wissenschaftliche Arbeit zum Vortrag strukturieren und schreiben
- können Inhalte eines gehörten Vortrags in einen Kontext einordnen und fachlich diskutieren.

Inhalte:

Im Proseminar Informatik werden jeder Studentin und jedem Studenten ein Thema, das auf einer grundlegenden Vorlesung des ersten Studienjahres aufbaut, zugewiesen. In der Ankündigung und in einer Vorbesprechung werden diese Themen vom Dozenten vorgestellt und die zugehörige Literatur genannt. Jeder Teilnehmer und jede Teilnehmerin wählt eines dieser Themen aus, erstellt dazu unter Anleitung eine wissenschaftliche Arbeit (ca 10 bis 15 Seiten) sowie einen dazugehörigen Foliensatz und hält einen etwa 30-minütigen Vortrag. In der anschließenden Diskussion geht es neben der fachlichen Erörterung auch um die Bewertung der Präsentation durch die Seminarteilnehmer.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Proseminar	2	Vortrag Teilnahme an den Diskussionen zum Vortrag	Präsenzzeit Vortragsvorbereitung Ausarbeitung	30 60 60
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Ja			
Arbeitszeitaufwand insgesamt	150 Stunden	5 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Semester			
Verwendbarkeit				

11 Hier bitte frei lassen.

Modul: Softwareprojekt	[12]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen: Ein passendes Vertiefungsmodul, Softwaretechnik	

Qualifikationsziele:

Die Studentinnen und Studenten können am Ende des Moduls:

- die in Softwaretechnilk erlernten Methoden praktisch umsetzen
- Qualitäts-, Aufwands-, Akzeptanz- und Erfolgskriterien verstehen
- mündliche und schriftliche Kommunikationstechniken zur Planung und Durchführung kleiner Softwareprojekte im Team anwenden
- Methoden des Projektmanagements und Konfigurationsmanagement anwenden

Inhalte:

Im Softwareprojekt wird von den Studierenden im Team unter Anleitung des Dozenten ein größeres Softwaresystem arbeitsteilig entwickelt. Dabei werden alle Phasen eines Softwareprojekts durchlaufen sowie typische Methoden und Hilfsmittel, wie sie im Modul Softwaretechnik kennen gelernt wurden, eingeübt . Dazu gehören:

- Definieren, Abstimmen und Dokumentieren von Schnittstellen
- Arbeitsteilige Erstellung von Softwarekomponeneten im Team unter Anleitung eines studentischen Tutors, dabei Verwenden noch nicht implementierter Schnittstellen
- Eine noch fremde Technologie oder größere Softwarekomponente selbständig beurteilen und erlernen (Wiederverwendung)
- Durchsichten von Anforderungen, Schnittstellen, Implementierungen, Testfällen
- Modultest, Integrationstest, Systemtest; einschließlich Automatisierung und Rückfalltesten
- Versions- und Konfigurationsverwaltung, Build-Prozesse und Werkzeuge

Den Studentinnen und Studenten wird empfohlen, dieses Modul unmittelbar im Anschluss an die Präsenzphase des Moduls „Softwaretechnik“ zu absolvieren.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Projekt	4	Bearbeiten der Teilaufgaben, Dokumentieren und Präsentieren der Ergebnisse	Präsenzzeit Praktikumsarbeit Prüfungsvorbereitungszeit	60 210 30

Veranstaltungssprache	Deutsch	
Pflicht zur regelmäßigen Teilnahme	Ja.	
Arbeitszeitaufwand insgesamt	300 Stunden	10 LP
Dauer des Moduls	1 Semester	
Häufigkeit des Angebots	Jedes Semester	
Verwendbarkeit		

12 Hier bitte frei lassen.

Modul: Logik und Diskrete Mathematik	[13]

Hochschule/Fachbereich/Institut: Freie Universität Berlin/

Modulverantwortliche/er:

Zugangsvoraussetzungen:

Qualifikationsziele:

Die Studentinnen und Studenten kennen am Ende des Moduls:

- Entscheidungsmethoden für Aussagenlogik
- die grundlegenden Konzepte der Logik, Mengenlehre und der Diskreten Mathematik
- verschiedene Beweistechniken

Die Studentinnen und Studenten können am Ende des Moduls:

- Sachverhalte abstrahieren, in einer Logik formalisieren und logische Ausdrücke verbalisieren
- einfache Sachverhalte mathematisch modellieren
- sind in der Lage, Beweise nachzuvollziehen bzw. einfache mathematische Beweise selbst zu führen.

Inhalte:

- Aussagenlogik und mathematische Beweistechniken

- Boolesche Formeln und Boolesche Funktionen, DNF und KNF, Erfüllbarkeit, Resolutionskalkül

- Mengenlehre: Mengen, Relationen, Äquivalenz- und Ordnungsrelationen, Funktionen

- Natürliche Zahlen und vollständige Induktion, Abzählbarkeit

- Prädikatenlogik und mathematische Strukturen

- Kombinatorik: Abzählprinzipien, Binomialkoeffizienten und Stirling-Zahlen, Rekursion, Schubfachprinzip

- Graphentheorie: Graphen und ihre Darstellungen, Wege und Kreise in Graphen, Bäume

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	4	-	Präsenzzeit Vorlesung Präsenzzeit Übung Sebststudium Prüfungsvorbereitung	60 30 120 30
Übung	2			
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung			
Arbeitszeitaufwand insgesamt	240 Stunden	8 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Wintersemester			
Verwendbarkeit				

13 Hier bitte frei lassen.

Modul: Lineare Algebra	[14]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen:	

Qualifikationsziele:

Die Studentinnen und Studenten kennen am Ende des Moduls:

- grundlegende Stukturen der linearen Algebra (Gruppe, Ring, Modul, Körper, Vektorraum)
- Methoden zur Problemlösung in der linearen Algebra

Die Studentinnen und Studenten können am Ende des Moduls:

- Anwendungsprobleme mathematisch zu beschreiben und das dahinter liegende mathematische Kernproblem formulieren.
- Erkennen, welche Methoden zur Problemlösung geeignet sind, und diese Anwenden
- Lösungsverfahren algorithmisch Umsetzen

Inhalte:

Prädikatenlogik, mathematische Beweistechniken, elementare Mengenlehre, Induktionsprinzipien

Lineare Algebra: Vektorraum, Basis und Dimension; lineare Abbildung, Matrix und Rang; Gauss-Elimination und lineare Gleichungssysteme; Ringe, Polynomringe, Determinanten, Eigenwerte und Eigenvektoren; Euklidische Vektorräume und Orthonormalisierung; Hauptachsentransformation

Anwendungen der linearen Algebra in der affinen Geometrie, Statistik und Codierungstheorie (lineare Codes)

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	4	-	Präsenzzeit Vorlesung Präsenzzeit Übung Selbststudium Prüfungsvorbereitung	60 30 120 30
Übung	2	Schriftliche Bearbeitung der Übungsblätter mündliche Präsentation der Lösungen von Übungsaufgaben in den Übungen		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung			
Arbeitszeitaufwand insgesamt	240 Stunden	8 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Wintersemester			
Verwendbarkeit				

14 Hier bitte frei lassen.

Modul: Analysis	15
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen:	

Qualifikationsziele:
Die Studentinnen und Studenten kennen am Ende des Moduls: Die Zahlbereiche von den natürlichen zu den komplexen Zahlen und die Probleme der Repräsentation in Rechnern; Konvergenz von Folgen, Reihen, Funktionen; Grundlegende Toplogie und Stetigkeit; Differentialrechnung, gewöhnliche und inhomogene Differentialgleichungen, Superpositionsprinzip; Grundbegriffe der Maßtheorie; Integration, Stammfunktion, Riemann- und Lebesgue-Integral; Grundlagen diskreter und stetiger Stochastik. Insbesondere kennen sie ausgewählte numerische Methoden zur Lösung von Problemen der Analysis.

Die Studentinnen und Studenten können am Ende des Moduls:

- Kentnisse zum tieferen Verständnis der Differential- und Integralrechnung einsetzen.
- geeignete Anwendungsprobleme mathematisch modellieren und mit den Mitteln der Differential- und Integralrechnung lösen.
- Einige numerische Standardmethoden anwenden.

Inhalte:
Prädikatenlogik, mathematische Beweistechniken, elementare Mengenlehre, Induktionsprinzipien; Aufbau der Zahlenbereiche von den natürlichen bis zu den reellen Zahlen, Vollständigkeitseigenschaft der reellen Zahlen; Polynome, Nullstellen und Polynominterpolation; Exponential- und Logarithmusfunktion, trigonometrische Funktionen; Komplexe Zahlen, komplexe Exponentialfunktion und komplexe Wurzeln; Konvergenz von Folgen und Reihen, Konvergenz und Stetigkeit von Funktionen; Differentialrechnung: Ableitung einer Funktion, ihre Interpretation und Anwendungen; numerische Methoden zum Lösen von Differentialgleichungen und Differentialgleichungssystemen; Intergralrechnung: Bestimmtes und unbestimmtes Integral, Hauptsatz der Differential- und Intergralrechnung, Anwendungen
Grundbegriffe der Stochastik: Diskrete und stetige Wahrscheinlichkeitsräume, Unabhängigkeit von Ereignissen; Zufallsvariable und Standardverteilungen; Erwartungswert und Varianz

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	4	-	Präsenzzeit Vorlesung Präsenzzeit Übung Selbststudium Prüfungsvorbereitung	60 30 120 30

15 Hier bitte frei lassen.

Übung	2	Schriftliche Bearbeitung der Übungsblätter mündliche Präsentation der Lösungen von Übungsaufgaben in den Übungen
Veranstaltungssprache	Deutsch	
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung	
Arbeitszeitaufwand insgesamt	240 Stunden	8 LP
Dauer des Moduls	1 Semester	
Häufigkeit des Angebots	Jedes Sommersemester	
Verwendbarkeit		

Modul: Softwaretechnik	[16]

Hochschule/Fachbereich/Institut: Freie Universität Berlin/

Modulverantwortliche/er:

Zugangsvoraussetzungen:

Qualifikationsziele:
Die Studentinnen und Studenten verstehen die wesentlichen Fragestellungen für die Entwicklung großer Systeme; verstehen die wesentlich unterschiedlichen Randbedingungen, unter denen diese Entwicklung erfolgen kann; verstehen die wichtigsten Ansätze, mit denen diese Fragestellungen gelöst werden, und können ihre Eigenschaften analysieren; können beurteilen, unter welchen Umständen welche Ansätze Erfolg versprechend sind; können die wichtigsten dieser Ansätze selbst durchführen; beherrschen die Methoden des Projektmanagements und verstehen Gender- und Diversityaspekte im Projektmanagement.

Inhalte:

In der Vorlesung werden Prinzipien, Methoden und Techniken für die Entwicklung großer Programmsysteme einschließlich einer Anleitung zum Projektmanagement vermittelt. Wichtige Einzelfertigkeiten werden in der begleitenden Übung konkret erprobt.
Die Teilnehmenden lernen Antworten u.a. auf folgende Fragen:

- Wie findet man heraus, welche Eigenschaften eine Software haben soll? (Anforderungsermittlung)
- Wie beschreibt man dann diese Eigenschaften? (Anforderungsbeschreibung)
- Was macht gute Software aus? (Qualitätsmerkmale)
- Wie strukturiert man die Software so, dass sie sich leicht bauen und flexibel verändern lässt? (Architektur, Entwurf)
- Wie deckt man Mängel in Software auf? (Analytische Qualitätssicherung)
- Wie beugt man Mängeln vor? (Konstruktive Qualitätssicherung)
- Wie organisiert man die Arbeit einer Softwareabteilung oder eines Softwareprojekts, um regelmäßig kostengünstige und hochwertige Resultate zu erzielen? (Projektmanagement, Prozessmanagement, Organisation)

Den Studentinnen und Studenten wird empfohlen, das Modul „Softwaretechnik“ und das Softwareprojekt/Berufspraktikum in demselben Semester zu absolvieren.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	4	-	Präsenzzeit Vorlesung	60
			Präsenzzeit Übung	15
			Selbststudium	105
			schriftliches Refarat	30

16 Hier bitte frei lassen.

Übung	2	Beteiligung an den Diskussionen in der Übung, Präsentation eigener Rechercheergebnisse
Veranstaltungssprache	Deutsch	
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung	
Arbeitszeitaufwand insgesamt	240 Stunden	8 LP
Dauer des Moduls	1 Semester	
Häufigkeit des Angebots	Jedes Wintersemester	
Verwendbarkeit		

Modul: Anwendungssysteme	[17]
Hochschule/Fachbereich/Institut: Freie Universität Berlin/	
Modulverantwortliche/er:	
Zugangsvoraussetzungen:	

Qualifikationsziele:

Die Studentinnen und Studenten

- verstehen den Unterschied zwischen Verfügungswissen und Orientierungswissen,

- lernen, beim Nachdenken über Informatiksysteme zu unterscheiden zwischen technischen Fragestellungen, Technikfolgenabschätzung und Technikfolgenbewertung,

- verstehen die Verantwortungsaspekte der Ingenieurtätigkeit,

- erlernen einige Aspekte der Technikfolgenabschätzung in bestimmten Informatik-Themenbereichen wie z.B. Sicherheit, Schutz der Privatsphäre.

- verstehen Gender- und Diversityaspekte von Anwendungen und der Anwendungsentwicklung

Inhalte:

Dieses Modul behandelt die Auswirkungen der Informatik. Nach grundlegenden Fragen (Konzept 'Verfügungswissen', Verantwortungsbegriff, Subjektivität von Techniksoziologie) werden konkret an Beispielen Technikfolgen in informatiklastigen Gebieten behandelt, z.B. die Sicherheit softwareintensiver technischer Systeme, der Schutz der Privatsphäre oder Auswirkungen der Computerisierung der Arbeitswelt.

Lehr- und Lernformen	**Präsenzstudium** (Semester-wochenstunden = SWS)	**Formen aktiver Teilnahme**	**Arbeitsaufwand** (Stunden)	
Vorlesung	2	-	Präsenzzeit Vorlesung Präsenzzeit Übung Selbststudium schriftliches Refarat	30 30 60 30
Übung	2	Beteiligung an den Diskussionen in der Übung, Präsentation eigener Rechercheergebnisse		
Veranstaltungssprache	Deutsch			
Pflicht zur regelmäßigen Teilnahme	Nur zur Übung			
Arbeitszeitaufwand insgesamt	150 Stunden	5 LP		
Dauer des Moduls	1 Semester			
Häufigkeit des Angebots	Jedes Wintersemester			
Verwendbarkeit				

17 Hier bitte frei lassen.

Studienverlaufsplan (Beginn im Wintersemester)

Semester	
1	Rechnerarchitektur V3+Ü2, 7LP
2	Betriebssysteme und Kom V3+Ü2, 7LP
3	Hardwarepraktikum P2, 5LP
4	
5	Veranstaltungen im Vertief
6	Bachelorarbeit 12LP
Summe	14 SWS, 24 LP

Studienverlaufsplan (Beginn im Sommersemester)

Semester	
1	Betriebssysteme und Kom V3+Ü2, 7LP
2	Rechnerarchitektur V3+Ü2, 7LP
3	
4	Hardwarepraktikum P2, 5LP
5	Nichtsequentielle und Netz V4+Ü2, 8LP
6	Bachelorarbeit 12LP
Summe	14 SWS, 24 LP