School of Engineering

Weiterbildungen im Bereich Informatik

– MAS Informatik
– DAS Informatik
– CAS Computer Science 1
– CAS Computer Science 2
– CAS Object Oriented Programming
– CAS Software Engineering
– CAS Angewandte IT-Sicherheit
– CAS Information Engineering

Impressum

Text: ZHAW School of Engineering
Druck: Druckerei Schellenbergdruck AG, Pfäffikon
Papier: Lessebo Smooth White,
FSC- und PEFC-zertifiziert, CO_2-neutral
Januar 2024 – 500

Einleitung

Anwendungen in der Informations- und Kommunikationstechnik (engl. Information and Communication Technology, kurz ICT) sind im beruflichen und privaten Umfeld allgegenwärtig. Die Veränderungen durch die Informatik und ihre Anwendungen sind nachhaltig. Wir lernen, lehren und arbeiten anders. Institutionen und Verwaltungen können nur mit einer funktionsfähigen ICT ihre Aufgaben erfüllen; die Wirtschaft ist ohne optimierte ICT nicht mehr konkurrenzfähig.

Die fortschreitende Digitalisierung wird mittel- bis langfristig zu weiteren Umbrüchen führen. Ständig entstehen neue Technologien und Anwendungsgebiete. Ebenso schnell ändern und erweitern sich die Berufsbilder in der ICT. Heute sind hochqualifizierte, kommunikative Fachkräfte gefragt, die im Team innovative Lösungen entwickeln.

Der Fachkräftebedarf in der ICT ist seit einigen Jahren sehr hoch. Mit der zunehmenden Digitalisierung der Gesellschaft, Wirtschaft und Industrie wird sich diese Nachfrage auch in den kommenden Jahren weiter verstärken. Die Digitalisierung bietet für Unternehmen grosses Potenzial. Um die digitale Transformation in den Unternehmen umzusetzen, werden gut ausgebildete ICT-Fachkräfte benötigt. Dabei werden die Wirtschaft und die Industrie neben erfahrenen ICT-Fachkräften auch weiterhin auf ICT-Quereinsteiger angewiesen sein, die ihr Informatik-Wissen mit Weiterbildungen und on the job erlangt haben.

Auf der Basis Ihrer Ingenieur- oder betriebswirtschaftlichen Grundausbildung erhalten Sie zusätzlich eine fundierte und breite Ausbildung in sämtlichen wichtigen Themenfeldern der Informatik. Und als erfahrene ICT-Fachkraft erweitern und vertiefen Sie Ihr Informatikwissen, um aktuell zu bleiben. Damit stellen Sie die Weichen für eine erfolgreiche Laufbahn in der ICT-Branche. Sie sind federführend involviert bei der Konzeption, Umsetzung und dem Betrieb von neuen Informatiksystemen und -produkten oder leiten anspruchsvolle Informatikprojekte.

MAS, DAS und CAS im Bereich Informatik

Die Weiterbildungsmöglichkeiten im Bereich Informatik sind auf die individuellen Bedürfnisse der Studierenden bzw. deren Arbeitgeber zugeschnitten. Das Angebot reicht von einzelnen Zertifikatslehrgängen (Certificate of Advanced Studies) bis hin zum Master of Advanced Studies (MAS) in Informatik.

Zielpublikum
Das Weiterbildungsangebot im Bereich Informatik richtet sich an
– ICT-Quereinsteiger, die eine fundierte und breite Ausbildung in Informatik absolvieren wollen;
– ICT-Fachkräfte mit mehrjähriger Berufserfahrung und Spezialisten, die sich im Bereich Informatik als Ganzes oder in einem spezifischen Thema der Informatik weiterbilden bzw. spezialisieren wollen.

Modularer Aufbau
Das Angebot im Bereich Informatik ist modular aufgebaut und besteht aus folgenden Certificates of Advanced Studies (CAS):
– CAS Computer Science 1
– CAS Computer Science 2
– CAS Object Oriented Programming
– CAS Software Engineering
– CAS Angewandte IT-Sicherheit
– CAS Information Engineering

Informationsveranstaltung
Sie können sich über folgenden Link zu einer der regelmässig stattfindenden Informationsveranstaltungen anmelden:
www.zhaw.ch/engineering/weiterbildung

Zulassung
Die Zulassung zu einem MAS, DAS oder CAS setzt grundsätzlich einen Hochschulabschluss (Fachhochschule, HTL, HWV, Uni, ETH) voraus. Es können aber auch Praktikerinnen und Praktiker mit vergleichbarer beruflicher Kompetenz zugelassen werden, wenn sich die Befähigung zur Teilnahme aus einem anderen Nachweis ergibt.

Anmeldung
Anmelden können Sie sich direkt online unter:
www.zhaw.ch/engineering/weiterbildung

Das modulare Angebot im Bereich Informatik

Abschluss-Modul		Masterarbeit 12 ECTS			
Wahlpflicht-CAS (4 aus 6)	Vertiefungsthemen	CAS Object Oriented Programming 12 ECTS	CAS Software Engineering 12 ECTS	CAS Angewandte IT-Sicherheit 12 ECTS	CAS Information Engineering 12 ECTS
	Grundlagen	CAS Computer Science 1 12 ECTS	CAS Computer Science 2 12 ECTS		

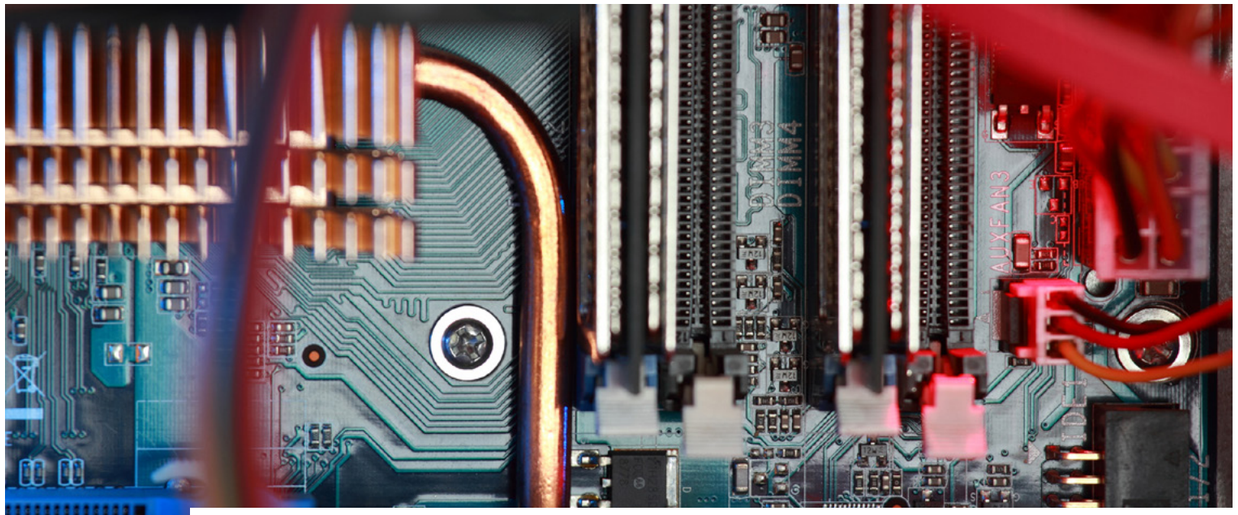
Die zwei Grundlagen-CAS Computer Science 1 und 2 vermitteln ICT-Quereinsteigern die fundamentalen Konzepte der theoretischen, technischen und praktischen Informatik (Kern-Informatik), die in den Vertiefungs-CAS vorausgesetzt werden. Die Vertiefungs-CAS vertiefen ein spezifisches Thema aus der praktischen oder angewandten Informatik.

Für die direkte Absolvierung von Vertiefungs-CAS müssen die jeweiligen Kenntnisse aus den zwei Grundlagen-CAS Computer Science 1 und 2 durch eine entsprechende vorgängige Ausbildung (in der Regel Hochschulabschluss oder höhere Berufsausbildung in Informatik) und einer mehrjährigen Berufserfahrung nachgewiesen werden können.

Um das DAS Informatik zu erlangen, müssen drei der oben aufgeführten Wahlpflicht-CAS absolviert werden. Um das Diplom des MAS Informatik zu erlangen, müssen vier der Wahlpflicht-CAS sowie das abschliessende Masterarbeitsmodul absolviert werden (vgl. Modulplan oben).

Studienleitung DAS, MAS Informatik
Kurt Bleisch
Telefon +41 58 934 41 55
kurt.bleisch@zhaw.ch

CAS Computer Science 1



Überblick
Der Computer ist in unserem heutigen Leben omnipräsent. Sei es als Bestandteil Ihres Notebooks, Handys, Fernsehers oder Ihrer Waschmaschine: In all diesen Geräten befinden sich Computer, die auf denselben Prinzipien eines digitalen Rechners funktionieren. In diesem CAS lernen Sie, wie ein Computer grundsätzlich funktioniert und wie Informationen in einem Computer codiert und verarbeitet werden. Weiter lernen Sie, wie Problemstellungen mit einem Algorithmus und einer Programmiersprache systematisch gelöst und wie grosse Informationsmengen strukturiert in relationalen Datenbanken verarbeitet werden.

Folgende Fragestellungen stehen im Zentrum des CAS Computer Science 1:
– Wie funktioniert ein Computer grundsätzlich?
– Wie wird Information im Computer codiert und verarbeitet?
– Wie sieht der Aufbau eines modernen Computers (Hard- und Software) aus?
– Wie programmiere ich einen Computer und was für Programmiersprachen gibt es?
– Wie entwerfe und programmiere ich systematisch einen Algorithmus zur Lösung eines Problems?
– Was ist eine Datenbank und wie kann ich Daten darin abfragen und bearbeiten?

Vorkenntnisse
Grundkenntnisse in Informatik und erste praktische Erfahrungen im ICT-Umfeld sind sinnvoll, aber nicht Voraussetzung. Weiter sollte Ihnen logisches und abstraktes Denken zur Problemlösung Freude machen.

Methodik
Das Ausbildungsprogramm umfasst verschiedene Aktivitäten, wie etwa Vorlesungen, praxisorientierte Übungen und Fallbeispiele, Gruppenarbeiten und Selbststudium (Vor- und Nachbereitung).

Unterrichtssprachen
Die Unterrichtssprache ist Deutsch. Fachliteratur und eingesetzte Softwareprogramme sind teilweise in Englisch.

Unterrichtszeiten
Der Unterricht findet berufsbegleitend jeweils am Dienstagnachmittag/-abend von 13.15 bis 20.45 Uhr (8 Lektionen) statt. Das CAS Computer Science 1 dauert rund fünf Monate. Den individuellen Stundenplan erhalten die Studierenden spätestens einen Monat vor Studienbeginn. Die schulfreie Zeit richtet sich nach den Schulferien der Stadt Zürich.
Bei grosser Nachfrage wird zusätzlich eine Durchführung am Donnerstag, von 9.00 bis 17.00 Uhr (8 Lektionen) angeboten.

Durchführungsort
ZHAW Zürcher Hochschule für Angewandte Wissenschaften, School of Engineering, Lagerstrasse 41, 8004 Zürich

Startdaten
Detaillierte Angaben zu den Startdaten finden Sie online unter www.zhaw.ch/engineering/weiterbildung

Studienleitung
Kurt Bleisch
Telefon +41 58 934 41 55
kurt.bleisch@zhaw.ch

Struktur und Inhalt

Modul	Inhalt	Lernziele	ECTS
Grundlagen Informatik	– Grundbegriffe wie Bit, Byte und Wort – Zahlensysteme und Konvertierungsalgorithmen: Binär-, Oktal-, Dezimal- und Hexadezimalsystem – Rechnen im Binärsystem: Addition, Subtraktion, Multiplikation, Division – Darstellung von Zeichen und Text (ASCII, Unicode) – Informationstheorie: Entropie, Redundanz und Anwendung in der Datenkompression und Fehlererkennung – Gesetze einer Booleschen Algebra – Logische Funktionen und Gatter – Kombinatorische und sequentielle Grundschaltungen (Multiplexer, Addierer, Schieberegister, Zähler) – Aufbau und Funktionsweise eines Computers (Von-Neumann-Architektur) – Moderne Rechnerarchitekturen und alternative Computermodelle (z.B. Quantencomputer)	Sie sind in der Lage: – Codes zur Darstellung von Zahlen und Text zu benutzen – die zentralen Konzepte der Informationstheorie nach Shannon und deren Anwendung zu beschreiben – die Boolesche Algebra für Programmierung und Schaltungsentwurf anzuwenden – ein digitales Gatter zu erklären und die Realisation einfacher Funktionen sowie das Aufstellen und Interpretieren von Wahrheitstabellen anzuwenden – den schematischen Aufbau eines Rechners (Von-Neumann-Architektur), das Zusammenspiel von Hardware und Software zu erklären – den Aufbau eines einfachen Befehlssatzes und damit erstellte Programme in Assemblersprache zu erläutern	3
Programmier-sprachen	– Programme mit Python erstellen – Variablen und Datentypen – Kontrollstrukturen und Logik (Sequenz, Selektion, Iteration) – Arrays und Collections – Funktionen, Module – Entwurf und Analyse von Algorithmen (Greedy, Divide and Conquer, Rekursion, Komplexität und O-Notation) – Lösen praktischer, einfacher Programmieraufgaben – Beschreibung von formalen Sprachen – Charakteristika und Vertreter der einzelnen Generationen von Programmiersprachen – Imperatives, objektorientiertes und deklaratives Programmierparadigma	Sie sind in der Lage: – die grundlegenden Konzepte einer (prozeduralen) Programmiersprache anzuwenden – für einfache algorithmische und datenstrukturorientierte Aufgabenstellungen Programme mit einer prozeduralen Programmiersprache zu entwickeln – Grundkonzepte der Beschreibung von formalen Sprachen in deklarativer Form oder mittels Grammatiken zu erläutern – die wichtigsten Charakteristika und Vertreter der einzelnen Generationen von Programmiersprachen aufzuzählen – Programmierparadigmen wie imperative, objektorientierte, funktionale und deklarative Programmierung zu unterscheiden	3
Hardwarenahe Programmierung	– Einführung in die Mikrocontroller-Programmierung mit C (Datentypen, Operatoren, Funktionen) – Analoge und digitale Ein- und Ausgabeschnittstellen (GPIO, I2C, SPI) – Einlesen von Sensorwerten (Temperatur, Luftfeuchtigkeit, Beschleunigung etc.) – Ansteuern von Aktoren (LED, Buzzer, LCD, Motoren etc.) – Anwendungen wie Internet of Things (IoT) – Praktisches Arbeiten mit einem Mikrocontroller (Arduino-Plattform) für einfache Überwachungs-, Steuerungs- und Regelungsaufgaben	Sie sind in der Lage: – den Aufbau und den Einsatz eines Mikrocontrollers für technische Anwendungen zu skizzieren – den Entwurf und die Integration einfacher Hardwarekomponenten in ein bestehendes (embedded) System zu verstehen und an einfachen Beispielen anzuwenden – die Nutzung einfacher Ein-/Ausgabeschnittstellen, z. B. zur Implementierung von Steuerungen/Regelungen anwenden zu können – einen Lösungsentwurf bestehend aus Hard- und Software für eine Projektidee zu erstellen, – die Projektidee systematisch umzusetzen und auszutesten	3
Datenbanken	– Einführung in Datenbanksysteme – Datenmodellierung (Entity-Relationship-Diagramm) und Normalformen – Relationale Algebra und Datenbankabfragesprache SQL – Datenbankprogrammierung (Stored Procedures, Trigger) – Aufbau und Zweck von Indexen – Transaktionen, ACID-Prinzip – Praktisches Arbeiten mit einem Datenbanksystem (z. B. MySQL) und Anbindung an eine Anwendung (Python) – Alternative Datenbank-Modelle (z. B. NoSQL)	Sie sind in der Lage: – die Grundkonzepte relationaler Datenbanksysteme zu erklären – konzeptionelle Datenbankentwürfe für eine einfache Datenbankanwendung zu erstellen und in ein korrekt normalisiertes relationales Datenbankschema zu überführen – effiziente und korrekte Datenbankabfragen in SQL zu formulieren – die Funktionsweise von gespeicherten Prozeduren und Triggern zu verstehen und können einfache Aufgabenstellungen in einer DB-Programmiersprache einsetzen – das Konzept der Transaktion und die ACID-Eigenschaften zu erläutern	3
	Total ECTS-Punkte		**12**

CAS Computer Science 2

Überblick
Dieses CAS baut auf dem CAS Computer Science 1 auf und vermittelt die Themen Betriebssysteme, Datenkommunikation und verteilte Systeme. Betriebssysteme sind komplexe, nebenläufige Softwaresysteme, die dem Programmierer einerseits eine geeignete, langlebige Abstraktion der zugrunde liegenden Hardware zur Verfügung stellen und andererseits die Betriebsmittel eines Rechners effektiv verwalten. Mit Datenkommunikation bezeichnet man grundsätzlich alle Methoden, die (Nutz-)Informationen von einem Sender zu einem Empfänger übermitteln. Verteilte Systeme sind allgegenwärtig im beruflichen und privaten Alltag (WhatsApp, Internet u. v. m.); sie führen Konzepte und Technologien aus unterschiedlichen Bereichen der Informatik zusammen. Viele verteilte Systeme werden heute in einer Cloud (IT-Infrastruktur, die über das Internet verfügbar ist) betrieben.

Folgende Fragestellungen stehen im Zentrum des CAS Computer Science 2:
- Wie funktioniert ein Betriebssystem und was sind die Unterschiede in gängigen Betriebssystemen (Windows, MacOS, Linux)?
- Wie werden Daten für eine Datenübertragung codiert und wie funktionieren Kommunikationsnetzwerke?
- Wie kann Information sicher übertragen werden?
- Wie sind verteilte Systeme und insbesondere internetbasierte Systeme aufgebaut?
- Was sind die Anwendungen von Cloud Computing?

Vorkenntnisse
Die im CAS Computer Science 1 aufgeführten Kompetenzen werden vorausgesetzt.

Methodik
Das Ausbildungsprogramm umfasst verschiedene Aktivitäten, wie etwa Vorlesungen, praxisorientierte Übungen und Fallbeispiele, Gruppenarbeiten und Selbststudium (Vor- und Nachbereitung).

Unterrichtssprachen
Die Unterrichtssprache ist Deutsch. Fachliteratur und eingesetzte Softwareprogramme sind teilweise in Englisch.

Unterrichtszeiten
Der Unterricht findet berufsbegleitend jeweils am Dienstagnachmittag/-abend von 13.15 bis 20.45 Uhr (8 Lektionen) statt. Das CAS Computer Science 2 dauert rund fünf Monate. Den individuellen Stundenplan erhalten die Studierenden spätestens einen Monat vor Studienbeginn. Die schulfreie Zeit richtet sich nach den Schulferien der Stadt Zürich.

Durchführungsort
ZHAW Zürcher Hochschule für Angewandte Wissenschaften, School of Engineering, Lagerstrasse 41, 8004 Zürich

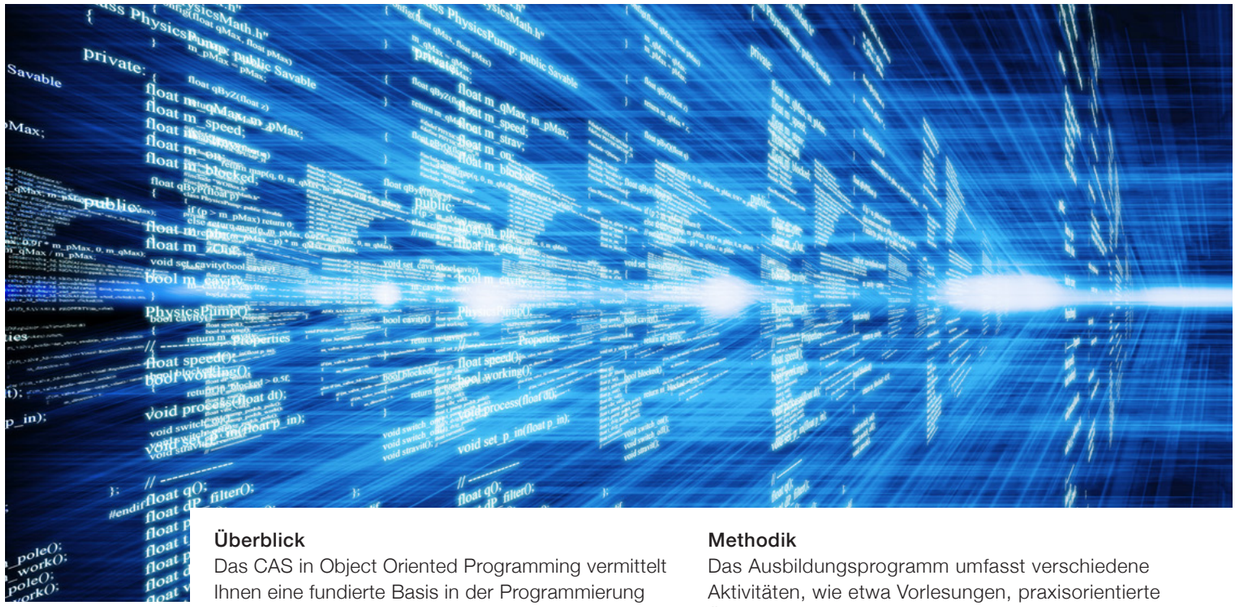
Startdaten
Detaillierte Angaben zu den Startdaten finden Sie online unter www.zhaw.ch/engineering/weiterbildung

Studienleitung
Kurt Bleisch
Telefon +41 58 934 41 55
kurt.bleisch@zhaw.ch

Struktur und Inhalt

Modul	Inhalt	Lernziele	ECTS
Betriebssysteme	– Einführung, Geschichte der Betriebssysteme, Arten und Einsatzbereiche von Betriebssystemen – Systemkonzepte und -strukturen – Prozesse und Threads – Scheduling – Prozesssynchronisation – Interprozesskommunikation – Speicherverwaltung, Virtual-Memory-Management – I/O-Management – Dateisysteme – Sicherheit und Schutzmechanismen – Beispiele, Fallstudien, Scripting mit Python	Sie sind in der Lage: – Aufgaben, Architektur, Funktionsweise und Schnittstellen (API) moderner Betriebssysteme zu erklären – die Mechanismen und Strategien von Betriebs- und Laufzeitsystemen zu beurteilen und zu bewerten – Dateisysteme und parallele Prozesse in Programmen zu nutzen – betriebssystemspezifische Programme zu erstellen (Shell-Skripts) – die Konzepte, die bei der Konstruktion von Betriebssystemen Verwendung finden, auf aktuelle Betriebssysteme zu übertragen – Sicherheitskonzepte von Betriebssystemen zu erläutern	4
Datenkommunikation	– Einführung in Übertragungstechniken, Quellen- und Kanalcodierung, Netzwerk-Topologie, Protokollspezifikation – Einführung in das ISO/OSI-Schichtenmodell (Referenzmodell) – Aktive Netzkomponenten wie Repeater, Bridge (Switch), Router, Gateways – Beispiele für Standardprotokolle: u. a. Ethernet, TCP/IP, DNS – Drahtlose Datenkommunikation: WLAN und Mobilfunknetze der 3. bis n-ten Generation sowie nahe Datenkommunikation (Bluetooth) – Netzwerkdesign – Fallstudien – Einführung in die Kryptologie zur sicheren Datenübertragung	Sie sind in der Lage: – die technischen Grundlagen der Datenkommunikation und von Computernetzwerken zu skizzieren – die Bedeutung von Schichtenmodellen und die Aufgaben und Funktionen der Schichten des ISO/OSI-Modells sowie die wichtigsten Dienstvertreter jeder Schicht zu erläutern – neue Netzwerk-Protokolle und Verfahren der Datenkommunikation selbstständig zu erarbeiten und zu verstehen – die speziellen Verfahren und die gängigen Protokolle der drahtlosen Datenkommunikation zu nennen – kleinere Computernetzwerke zu konzeptionieren und zu analysieren – die prinzipiellen Verfahren der Datenverschlüsselung zu skizzieren	4
Verteilte Systeme	– Einführung in verteilte Systeme – Internet und Web-basierte verteilte Systeme – Kommunikationsarten: Nachrichten-basierte (TCP/IP-Sockets), Web-basierte (Webservices mit REST) und Messaging-basierte (MOM) Kommunikation – Verteilte Algorithmen und Dienste (Zeitsynchronisation, verteilte Transaktionen, Authentifizierung und Autorisierung) – Anwenden und erstellen von einfachen verteilten Softwaresystemen mit Python – Virtualisierungstechnologie: Verfahren für CPU-, Memory-, Netzwerk- und Disk-Virtualisierung – Virtuelle Massenspeichersysteme – Einführung in Cloud Computing – Servicemodelle – Organisatorische Arten von Clouds	Sie sind in der Lage: – die charakteristischen Eigenschaften von verteilten Systemen und die grundlegenden Fragestellungen beim Entwurf von verteilten Systemen zu nennen – die Funktionsweise des Internets im Kern und in den Endsystemen (Webbrowser und Webserver) zu erklären – die Funktionsweise der wichtigsten Kommunikationsarten (als Middleware implementiert) zum Bau von verteilten Systemen zu erklären und anzuwenden – Vorgänge in verteilten Systemen nachzuvollziehen, zu protokollieren und eine Fehlersuche durchzuführen – den Aufbau, den Zweck und die Einsatzmöglichkeiten der verschiedenen virtuellen Systeme zu erläutern – einfache Virtualisierungsinfrastrukturen zu verstehen und selbst aufzubauen; – Potenzial und Einsatzmöglichkeiten von Cloud Computing einzuschätzen	4
	Total ECTS-Punkte		**12**

CAS Object Oriented Programming



Überblick
Das CAS in Object Oriented Programming vermittelt Ihnen eine fundierte Basis in der Programmierung von modernen Informatik-Applikationen. Dabei lernen Sie die Grundlagen und Konzepte der objektorientierten Softwareentwicklung kennen. Der Fokus wird dabei auf die Programmiersprachen Java und C# gesetzt, die seit langer Zeit in der Wirtschaft, der Technik und bei Behörden im Einsatz sind. Die erworbenen Programmierkenntnisse werden angewendet, um moderne Desktop-, Web- und Mobile-Applications zu entwickeln. Im Modul «Grundlagen objektorientierter Programmierung» wird Java als Programmiersprache eingesetzt. Das Modul «Vertiefung objektorientierte Programmierung» kann wahlweise mit Java oder mit C# absolviert werden.

Folgende Fragestellungen stehen im Zentrum des CAS Object Oriented Programming:
- Wie werden Problemstellungen objektorientiert analysiert und programmiert?
- Wie werden grössere Programme systematisch entworfen und realisiert?
- Was sind die Technologien, um Web-Applikationen zu entwickeln und wie werden sie eingesetzt?
- Was für mobile Plattformen gibt es und wie werden Apps für eine mobile Plattform programmiert?

Vorkenntnisse
Grundlagen in Informatik und Kenntnisse in der Programmierung mit einer beliebigen höheren Programmiersprache (mindestens 3 ECTS oder vergleichbare Praxis) werden vorausgesetzt.

Methodik
Das Ausbildungsprogramm umfasst verschiedene Aktivitäten, wie etwa Vorlesungen, praxisorientierte Übungen und Fallbeispiele, Gruppenarbeiten und Selbststudium (Vor- und Nachbereitung).

Unterrichtssprachen
Die Unterrichtssprache ist Deutsch. Fachliteratur und eingesetzte Softwareprogramme sind teilweise in Englisch.

Unterrichtszeiten
Der Unterricht findet berufsbegleitend jeweils am Freitagnachmittag/-abend von 13.15 bis 20.45 Uhr (8 Lektionen) statt. Das CAS Object Oriented Programming dauert rund fünf Monate. Den individuellen Stundenplan erhalten die Studierenden spätestens einen Monat vor Studienbeginn. Die schulfreie Zeit richtet sich nach den Schulferien der Stadt Zürich.

Durchführungsort
ZHAW Zürcher Hochschule für Angewandte Wissenschaften, School of Engineering, Lagerstrasse 41, 8004 Zürich

Startdaten
Detaillierte Angaben zu den Startdaten finden Sie online unter www.zhaw.ch/engineering/weiterbildung

Studienleitung
Kurt Bleisch
Telefon +41 58 934 41 55
kurt.bleisch@zhaw.ch

Struktur und Inhalt

Modul	Inhalt	Lernziele	ECTS
Grundlagen objektorientierter Programmierung	– Konzepte der objektorientierten Programmierung – Klassen, Objekte – Objektinteraktion, Collections (ArrayList, HashMap, HashSet) – Vererbung, Komposition, Polymorphie, Abstrakte Klassen, Interfaces – Klassenentwurf (lose Kopplung, Kohäsion, Kapselung) – Darstellung Design mit UML-Klassendiagramm – Softwarequalitätssicherung (Unit-Test, Dokumentation) – Clean Code, Code Smells und Refactoring	Sie sind in der Lage: – die objektorientierten Konzepte anhand der Programmiersprache Java zu erklären – wichtige Aspekte von Java zu erläutern und die Grundlagen der Programmiersprache Java-adäquat anzuwenden – Problemstellungen objektorientiert zu analysieren, Lösungen zu entwerfen und diese umzusetzen	4
Vertiefung objektorientierte Programmierung	– Ausgewählte Konzepte und Themen, zum Beispiel: – Exception Handling – Lambda-Ausdrücke – Innere Klassen / Anonyme Klassen – File Input / Output – Serialisierung und Deserialisierung – Netzwerkprogrammierung – Threads – Generics – Grundlagen XML, JSON – Grafische Desktopapplikationen – Datenbankanbindung (OR-Mapper,) – Anwenden und vernetzen des Gelernten in einer Projektarbeit	Sie sind in der Lage: – weitergehende Sprachelemente und Konzepte der Programmiersprache Java oder C# und deren Anwendung zu erläutern – die neuen Sprachelemente und Konzepte anhand von praktischen Beispielen und mit Übungen anzuwenden – sich selbständig weiteres, vertieftes Wissen zur Entwicklung von Applikationen mit Java anzueignen – eine komplexere Problemstellung systematisch objektorientiert zu entwerfen und umzusetzen	4
Web und Mobile Applications	– Einführung in HTML, CSS und JavaScript – Entwicklung von Webapplikationen (z. B. mit Node.js) – Document Object Model (DOM) und Ereignisbehandlung im Browser – Asynchrone Client-Server-Kommunikation (Ajax, Fetch-API) – Zustand (Cookies, Sessions) und Authentisierung – Entwurf und Implementierung einer Webapplikation mit einem Web-Framework (z. B. Angular) – Überblick über Mobilplattformen – Mobilplattformen, Web und native Apps, Appstores – Applikationsdesign für Mobilanwendungen – Usability-Kriterien und -Tests	Sie sind in der Lage: – die grundlegenden Konzepte und Sprachen im Webumfeld (HTML, CSS, JavaScript) einzuordnen – die wesentlichen Merkmale und Unterschiede der client- und serverseitigen Programmierung zu diskutieren – selbständig kleine clientseitige, dynamische Webanwendungen zu erstellen – die wichtigsten Mobilplattformen und die eingesetzten Gerätetypen zu diskutieren – die speziellen Eigenschaften mobiler Benutzerschnittstellen und weitere wichtige Aspekte, die beim Design und bei der Entwicklung mobiler Applikationen zu berücksichtigen sind, auf eigene Anwendungen zu übertragen	4
	Total ECTS-Punkte		**12**

CAS Software Engineering

Überblick
Das CAS Software Engineering vermittelt Ihnen die Grundlagen, um ein Informatiksystem oder -produkt systematisch, d. h. unter Verwendung von bewährten Prinzipien, Methoden und Werkzeugen, zu realisieren.

Folgende Fragestellungen stehen im Zentrum des CAS Software Engineering:
– Wie müssen Projektteams und der Softwareprozess organisiert werden, um effizient und effektiv Software entwickeln zu können?
– Wie werden Anforderungen erhoben, verhandelt und beschrieben?
– Wie werden eine zweckmässige und wartbare Softwarearchitektur und ein Design erstellt?
– Wie kann sichergestellt werden, dass das Softwareprodukt die geforderte Qualität aufweist?

Vorkenntnisse
Grundkenntnisse in Informatik und einer höheren, objektorientierten Programmiersprache (z. B. Java, C#, C/C++) werden vorausgesetzt. Weiter sind mindestens ein bis zwei Jahre Erfahrung in Informatikprojekten sinnvoll, um die Theorie verstehen und festigen zu können.

Methodik
Das Ausbildungsprogramm umfasst verschiedene Aktivitäten, wie etwa Vorlesungen, praxisorientierte Übungen und Fallbeispiele, Gruppenarbeiten und Selbststudium (Vor- und Nachbereitung).

Unterrichtssprachen
Die Unterrichtssprache ist Deutsch. Fachliteratur und eingesetzte Softwareprogramme sind teilweise in Englisch.

Unterrichtszeiten
Der Unterricht findet berufsbegleitend jeweils am Donnerstagnachmittag/-abend von 13.15 bis 20.45 Uhr (8 Lektionen) statt. Das CAS Software Engineering dauert rund fünf Monate. Den individuellen Stundenplan erhalten die Studierenden spätestens einen Monat vor Studienbeginn. Die schulfreie Zeit richtet sich nach den Schulferien der Stadt Zürich.

Durchführungsort
ZHAW Zürcher Hochschule für Angewandte Wissenschaften, School of Engineering, Lagerstrasse 41, 8004 Zürich

Startdaten
Detaillierte Angaben zu den Startdaten finden Sie online unter www.zhaw.ch/engineering/weiterbildung

Studienleitung
Kurt Bleisch
Telefon +41 58 934 41 55
kurt.bleisch@zhaw.ch

Struktur und Inhalt

Modul	Inhalt	Lernziele	ECTS
Menschen, Prozesse und Requirements Engineering	– Einführung ins Software Engineering und in Softwareentwicklungsprozesse – Kriterien und Vorgehen für das Tailoring eines Softwareentwicklungsprozesses – Agile Softwareentwicklung mit Scrum – Einführung ins Requirements Engineering – Ermittlungstechniken (Befragungs-, Kreativitäts-, Beobachtungs- und unterstützende Techniken) – Anforderungen natürlichsprachlich und modellbasiert mit der UML kommunizieren und dokumentieren (Uses Cases, Domänenmodellierung) – Dokumentation im agilen Umfeld (Scrum) – Prüfen, abstimmen und verwalten (Qualitätsaspekte, Prinzipien und Techniken) – Strategien für Outsourcing und Offshoring – Prozesse, Rollen, Verträge und Qualitätskontrolle für verteilte Entwicklung und Outsourcing – Werkzeugunterstützung für verteilte Entwicklung und Outsourcing	Sie sind in der Lage: – die Charakteristiken der gängigen Softwareentwicklungsprozesse (plangetrieben, agil) und deren Anwendungsgebiete zu skizzieren – Kriterien für mehr oder weniger Zeremonie in einem Softwareprozess (Tailoring) zu nennen; – Softwareprojekte zu planen, durchzuführen und zu überwachen – die Bedeutung des Requirements Engineering für den Projekterfolg zu erläutern – situationsgerecht Techniken zur Ermittlung, Dokumentation und Konsolidierung von Anforderungen auszuwählen und anzuwenden – strategische Entscheide für Software-Sourcing vorzubereiten – die Risiken und Erfolgsfaktoren in der verteilten Entwicklung zu nennen – Zulieferer für Software zu bewerten – verteilte Projekte aufzusetzen und zu führen	5
Softwarearchitektur und -design	– Grundlagen von Softwarearchitekturen (Tätigkeiten, grundlegende Konzepte, Typen von softwareintensiven Systemen) – Entwurf von Architekturen (Vorgehens-, Architektur- und Design Patterns, Domain-Driven-Design, Clean Architecture) – Use Case Realisierung und Klassendesign (Responsability Driven Design) – Modellierung, Beschreibung und Kommunikation von Softwarearchitekturen mit der UML – Qualität und Bewertung von Softwarearchitekturen (Werkzeuge für Softwarearchitekten (UML Tools, Architekturanalyse-Tools etc.)	Sie sind in der Lage: – Anforderungen in ein (objektorientiertes) Softwaredesign zu überführen und zu implementieren – gängige Techniken zum Entwurf, der Beschreibung und Kommunikation von Softwarearchitekturen anzuwenden – Qualitätskriterien zu benennen und eine Softwarearchitektur zu bewerten	4
Softwaretest	– Grundlagen und Begrifflichkeit des Softwaretestens – Testprozess, Teststrategien und -Stufen (Unit-, Integrations-, System- und Akzeptanztests) – Testing im agilen Umfeld (Scrum) – Überblick und Einsatz von Test-Techniken für statische und dynamischer Tests – Einführung ins Test-Management (Organisation, Planung, Strategie, Wirtschaftlichkeitsaspekte) – Testautomatisierung (für unterschiedliche Anwendungen und die höheren Teststufen wie z. B. Systemtest)	Sie sind in der Lage: – Tests für Ihre erstellte Software systematisch zu planen und durchzuführen – situationsgerecht Testtechniken für die verschiedenen Teststufen (Unit-, Integrations-, Systemtest und Akzeptanztest) auszuwählen und anzuwenden – die Grundkonzepte des Test-Managements zu erläutern	3
	Total ECTS-Punkte		**12**

CAS Angewandte IT-Sicherheit

Überblick
Das CAS Angewandte IT-Sicherheit richtet sich gleichermassen an Informatikerinnen und Informatiker sowie Praktikerinnen und Praktiker aus verwandten Fachbereichen.
- Sie haben die Chance, Ihre Fähigkeiten im Bereich IT-Security und somit in einem der grössten Wachstumsmärkte der IT auf- bzw. auszubauen.
- Das Angebot bietet den Studierenden ein solides technisches Rüstzeug sowie Grundkenntnisse in verwandten Bereichen wie Sicherheits- und Risikomanagement sowie Cybersicherheit an.
- Während der Schwerpunkt im technischen Bereich liegt, werden deren rechtliche und betriebliche Aspekte ebenfalls behandelt.
- Die Teilnehmenden werden in die Bereiche Sicherheitsarchitektur und -management, Kryptologie und Netzwerksicherheit sowie Software- und Systemsicherheit eingeführt.
- Sie erhalten in ausführlichen und praxisnahen Übungen die Gelegenheit, ihr neu erworbenes Wissen anzuwenden.

Vorkenntnisse
Grundkenntnisse in der Entwicklung und im Betrieb von IT-Systemen – nicht zwingend im Security-Bereich – sowie Grundkenntnisse der Programmierung in Java oder einer verwandten Programmiersprache werden vorausgesetzt.

Unterrichtssprachen
Die Unterrichtssprache ist Deutsch. Fachliteratur und eingesetzte Softwareprogramme sind teilweise in Englisch.

Methodik
Das Programm umfasst verschiedene Aktivitäten, einschliesslich Vorlesungen, praxisorientierter Fallbeispiele, Gruppenarbeiten sowie Selbststudium zur Vor- und Nachbereitung des Unterrichts. Die Studierenden erhalten die Möglichkeit, die in der Vorlesung erarbeiteten Konzepte in vorlesungsbegleitenden Übungen bzw. Praktika anzuwenden.

Unterrichtszeiten
Der Unterricht findet berufsbegleitend einmal pro Woche jeweils am Dienstag von 9 bis 17 Uhr (8 Lektionen) statt. Das CAS Angewandte IT-Sicherheit dauert rund 16 Wochen. Den individuellen Stundenplan erhalten die Studierenden spätestens einen Monat vor Studienbeginn. Die schulfreie Zeit richtet sich nach den Schulferien der Stadt Zürich.

Durchführungsort
ZHAW Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
Lagerstrasse 41
8004 Zürich

Startdaten
Detaillierte Angaben zu den Startdaten finden Sie online unter www.zhaw.ch/engineering/weiterbildung

Studienleitung
Dr. Martin Ochoa Ronderos
Telefon +41 58 934 63 34
martin.ochoaronderos@zhaw.ch

Die im CAS Angewandte IT-Sicherheit vermittelten Kompetenzen orientieren sich an den Fähigkeitsprofilen für IT- und Informationssicherheitsverantwortliche. Das CAS Angewandte IT-Sicherheit ist modular aufgebaut und besteht aus drei Modulen.

Struktur und Inhalt

Modul	Inhalt	Lernziele	ECTS
Security Architecture and Management	– Sicherheitsgrundlagen – Sicherheitsarchitektur – Sicherheitsmanagement	– Die Studierenden kennen die organisatorischen, führungstechnischen und betrieblichen Grundlagen der IT-Sicherheit. – Sie sind in der Lage, ein Sicherheitskonzept zu entwerfen und bei der Umsetzung mitzuwirken. – Sie beherrschen die Grundlagen der einschlägigen Standards und Best Practices und können diese in der Praxis anwenden.	3
Cryptography and Network Security	– Kryptologie – Methoden und Protokolle zur Sicherung von Systemen im Internet – Angriffs- und Verteidigungstechniken	– Die Studierenden verstehen die Grundlagen der Kryptologie (Algorithmen, Einsatzzwecke, Resilienz), die die Basis für eine Vielzahl moderner Sicherheitsmechanismen bilden. – Sie verstehen moderne Protokolle und Methoden, um die Netzwerkkommunikation sowie den Zugang zu Systemen und Anwendungen zu sichern. Sie können Protokolle und Methoden zweckmässig anwenden und verstehen deren Möglichkeiten und Grenzen. – Sie kennen Angriffstechniken gegen Netzwerke und Systeme und können Angriffe nachvollziehen.	4
Software and Systems Security	– Sicherer Softwareentwicklungsprozess – Sicheres Programmieren (Fokus Java) – Security Testing	– Die Studierenden können die Prinzipien der sicheren Softwareentwicklung auf einen beliebigen Softwareentwicklungsprozess anwenden. Sie kennen typische sicherheitsrelevante Programmierfehler und wissen, wie sie diese identifizieren und verhindern können. – Sie können sichere Applikationen am Beispiel von Java entwickeln und setzen dabei die bestehenden Sicherheitsfeatures und Security Libraries zweckmässig ein. – Sie können Applikationen durch Verwendung geeigneter Methoden und Tools hinsichtlich Sicherheit testen und gefundene Schwachstellen auch ausnutzen.	5
Total			**12**

Für die abzudeckenden Fähigkeiten im IT- und IT-Sicherheitsbereich existieren seit einiger Zeit Frameworks aufbauend auf dem europäischen e-Kompetenz-Rahmenwerk (e-CF), in der Schweiz kodifiziert als SN 16234 (e-Kompetenz-Rahmen [e-CF] – ein gemeinsamer europäischer Rahmen für IKT-Fach- und Führungskräfte in allen Branchen) sowie ISO/IEC 27021 (Competence requirements for information security management systems professionals). Die Lerninhalte orientieren sich für die abgedeckten Bereiche an o. a. Normen.

CAS Information Engineering

Überblick
Die Sammlung, Aufbereitung und Nutzbarmachung von Informationen und Daten werden zunehmend zentraler. Unter Information Engineering verstehen wir Methoden und Verfahren zur Gestaltung und Entwicklung von Informationssystemen.

In diesem CAS lernen Sie, wie man sowohl mit strukturierten Daten (z. B. aus Datenbanken und Data Warehouses) als auch mit semistrukturierten und unstrukturierten Daten (z. B. Weblogs, Textdokumenten, Bildern, Videos etc.) umgeht.

Folgende Fragestellungen stehen im Zentrum des CAS Information Engineering:
- Welche Scripting-Methoden eignen sich für die Prozessierung von Daten?
- Was sind die Grundlagen einer relationalen Datenbank und wie kann ich Daten mit einer geeigneten Abfragesprache (SQL) filtern?
- Warum braucht man ein Data Warehouse und wie integriert man Daten aus unterschiedlichen Systemen?
- Was verbirgt sich hinter Big Data (Hadoop, Spark etc.) und welche neuen Fragestellungen lassen sich damit beantworten?
- Wie kann ich Sentimentanalyse für meine Unternehmung einsetzen, um neue Erkenntnisse über die Kundenzufriedenheit zu gewinnen und effektiv darauf zu reagieren?

Methodik
Das Ausbildungsprogramm umfasst verschiedene Aktivitäten, wie etwa Vorlesungen, praxisorientierte Übungen und Fallbeispiele, Gruppenarbeiten und Selbststudium (Vor- und Nachbereitung).

Zulassung
Die Zulassung zum CAS Information Engineering setzt Grundkenntnisse der Programmierung voraus.

Unterrichtssprachen
Die Unterrichtssprache ist Deutsch. Fachliteratur und eingesetzte Softwareprogramme sind teilweise in Englisch.

Unterrichtszeiten
Der Unterricht findet berufsbegleitend einmal pro Woche jeweils am Montag von 9 bis 17 Uhr (8 Lektionen) statt. Das CAS Information Engineering dauert rund fünf Monate. Den individuellen Stundenplan erhalten die Studierenden spätestens einen Monat vor Studienbeginn. Die schulfreie Zeit richtet sich nach den Schulferien der Stadt Winterthur.

Durchführungsort
ZHAW Zürcher Hochschule
für Angewandte Wissenschaften
School of Engineering
Technikumstrasse 9
8401 Winterthur

Startdaten
Detaillierte Angaben zu den Startdaten finden Sie online unter www.zhaw.ch/engineering/weiterbildung

Studienleitung
Dr. Andreas Weiler
Telefon +41 58 934 41 39
andreas.weiler@zhaw.ch

Struktur und Inhalt

Modul	Inhalt	Lernziele	ECTS
Scripting	– Einführung in Python mit sciPy und scikit-learn – Anwendungsmöglichkeiten in den Bereichen Datenextraktion, Datenanalyse und Datenvisualisierung – Erstellung von Mashups mit externen Web-Services	– Sie kennen die Grundlagen der Script-Sprache Python sowie der relevanten Bibliotheken. – Sie können die Script-Sprache für unterschiedliche Schritte im Datenanalyseprozess einsetzen.	3
Datenbanken und Data Warehousing	– Relationale Algebra und Datenbankabfragesprache SQL – Einführung in Decision-Support-Systeme: Definition, Abgrenzung, Vergleich OLTP (transaktionsbasierte Systeme) und OLAP (Analysesysteme) – Architektur und Modellierung: DWH-Aufbau, Datenmodellierung für Analysezwecke – ETL-Prozess: Kopplung von OLTP und Business-Intelligence(BI)-Welt, automatisiertes Laden, Datenqualität: Fehlererkennung und -korrektur, iteratives Vorgehen beim DWH-Entwurf	*Wie man strukturierte Daten aufbereitet, modelliert und für die Analyse bereitstellt.* – Sie verstehen die Grundlagen der relationalen Algebra und können die Datenbanksprache SQL anwenden. – Sie verstehen die Wesensmerkmale und den Aufbau sowie den Zweck von DWH-Systemen. – Sie können Architektur und Design von skalierenden DWH-Systemen entwerfen. – Sie kennen die Technologien und Bausteine von DWH-Systemen und sind in der Lage, diese Bausteine beispielhaft zur Implementation zu nutzen.	3
Information Retrieval	– Einführung in Information Retrieval – Grundlagen: Modelle, Probability Ranking Principle, Rangierungsregeln – Indizierung/Vergleich: Textanalyse, Gewichtung, Systeme/Architektur – Sentiment-Analyse, Text Summarization, mehrsprachiges und sprachübergreifendes Retrieval – Multimedia Information Access	*Wie man unstrukturierte Texte aufbereitet und nutzbar macht.* – Sie kennen konkrete Retrievalsysteme (z. B. Websuche/Google, fachspezifische Suche u. a.) und haben einen soliden Einblick in das Gebiet: Grundlagen, Theorie, Stand der Technik, Praxis und Auswertung. – Sie beherrschen die Wahl der richtigen Technologie für Suchaufgaben und können Information-Retrieval-Systeme evaluieren und bewerten. – Sie kennen Methoden der tiefergehenden Textanalyse wie Sentimentanalyse und können mit maschineller Übersetzung umgehen. – Sie lernen Methoden kennen, um Merkmale aus nicht textuellen Dokumenten zu extrahieren.	3
Big Data	– Big-Data-Überblick: Einsatzkonzepte für grosse und unstrukturierte Daten – Überblick über NoSQL – Skalierbare Abfragen und Analysen: MapReduce mit Hadoop, SQL-ähnliche Interfaces – Einführung in Apache Spark	*Wie man skalierbare Analysesysteme mit Big-Data-Technologie aufbaut und nutzt.* – Sie verstehen die Wesensmerkmale und den Aufbau sowie den Zweck von Big-Data-Systemen. – Sie können Big-Data-Systeme beurteilen und evaluieren. – Sie sind in der Lage, ein Big-Data-Projekt mit beliebiger Datenmenge durchzuführen. – Sie haben in den Praktika Hands-on-Erfahrung mit State-of-the-Art-Tools wie Apache Hadoop Ecosystem gesammelt.	3
	Total		**12**

Masterarbeit

Die Masterarbeit bildet einen zentralen Bestandteil des MAS-Studienganges und wird nach Abschluss aller notwendigen CAS erstellt. In der Masterarbeit wird das Gelernte an einer komplexeren Problemstellung aus der Praxis angewendet.

Überblick
Die Masterarbeit bildet den Abschluss des MAS in Informatik. Die Masterarbeit hat folgende Ziele.

Der Studierende ist in der Lage,
- ein berufsbezogenes Themengebiet zu bestimmen und zu begrenzen;
- klare Vorstellungen über das zu erreichende Ziel der Masterarbeit zu entwickeln;
- das Masterarbeitsthema eigenständig, nach wissenschaftlichen Kriterien systematisch und methodisch korrekt zu bearbeiten;
- seine Überlegungen durch logische Argumentation und eigenständige Interpretation nachvollziehbar zu machen;
- die Resultate der Masterarbeit auszuwerten und anderen zugänglich zu machen;
- die Ergebnisse formal korrekt zu präsentieren und in einem Diskurs begründen zu können.

Das Thema kann durch die Studierenden frei gewählt werden. Die Themenwahl soll sich aber an den Ausbildungsschwerpunkten der absolvierten CAS orientieren.

Zulassung
Studierende werden zur Masterarbeit zugelassen, wenn vier Wahlpflicht-CAS gemäss Modulplan bestanden und demzufolge 48 Credits erworben sind.

Ablauf
Die Studienleitung informiert vorgängig über Ziele und Ablauf der Masterarbeit. Weiter findet vor dem Start der Masterarbeit ein halbtägiger Workshop zum Thema wissenschaftliches Arbeiten statt.

Für die Anmeldung zur Masterarbeit erstellen die Studierenden eine Disposition, in der die Aufgabenstellung, Zielsetzung und Abgrenzung der Masterarbeit beschrieben wird.

Für die Bearbeitung der Masterarbeit wird den Studierenden ein Betreuer zugeteilt. In der Regel ist der Betreuer ein Dozierender aus einem der absolvierten CAS.

Die Masterarbeit wird neben den erarbeiteten Artefakten mit einem schriftlichen Bericht und einer Präsentation abgeschlossen.

Sprache
Die Masterarbeit kann in Deutsch oder Englisch erstellt und präsentiert werden.

Dauer
Die Masterarbeit dauert 4 Monate und umfasst einen Arbeitsaufwand von 12 ECTS (360 Std.). Die mündliche Präsentation der Masterarbeit findet nach Abschluss der Masterarbeit statt.

Startdaten
Jeweils im Frühjahr und im Herbst kann mit der Masterarbeit begonnen werden. Die konkreten Daten können der Website entnommen werden.

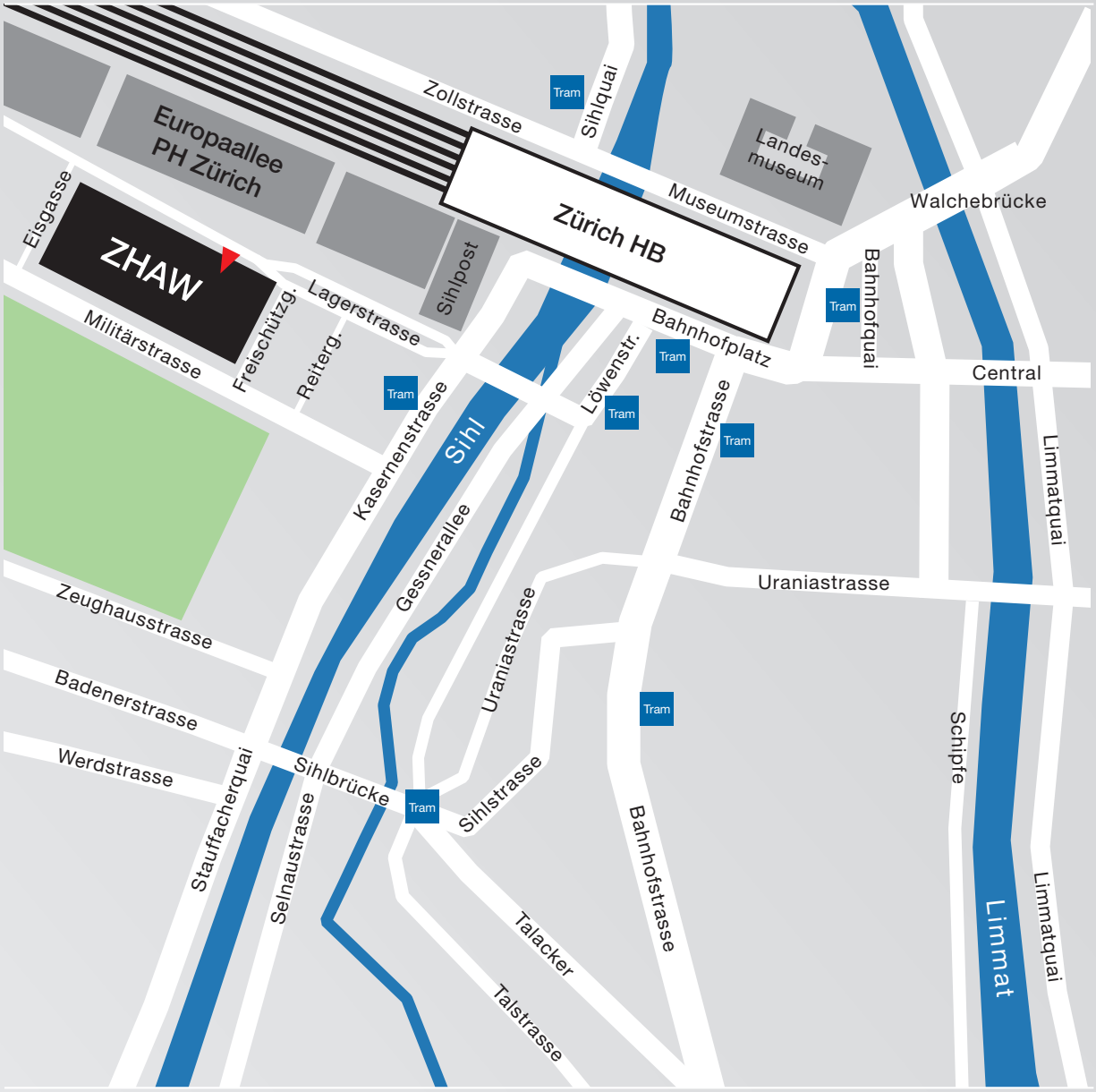
Studienleitung
Kurt Bleisch
Telefon +41 58 934 41 55
kurt.bleisch@zhaw.ch

Zürcher Hochschule
für Angewandte Wissenschaften

School of Engineering

Administration Weiterbildung Zürich
Lagerstrasse 41, Postfach
CH-8021 Zürich

Telefon +41 58 934 82 44
weiterbildung.engineering@zhaw.ch



www.zhaw.ch/engineering/weiterbildung