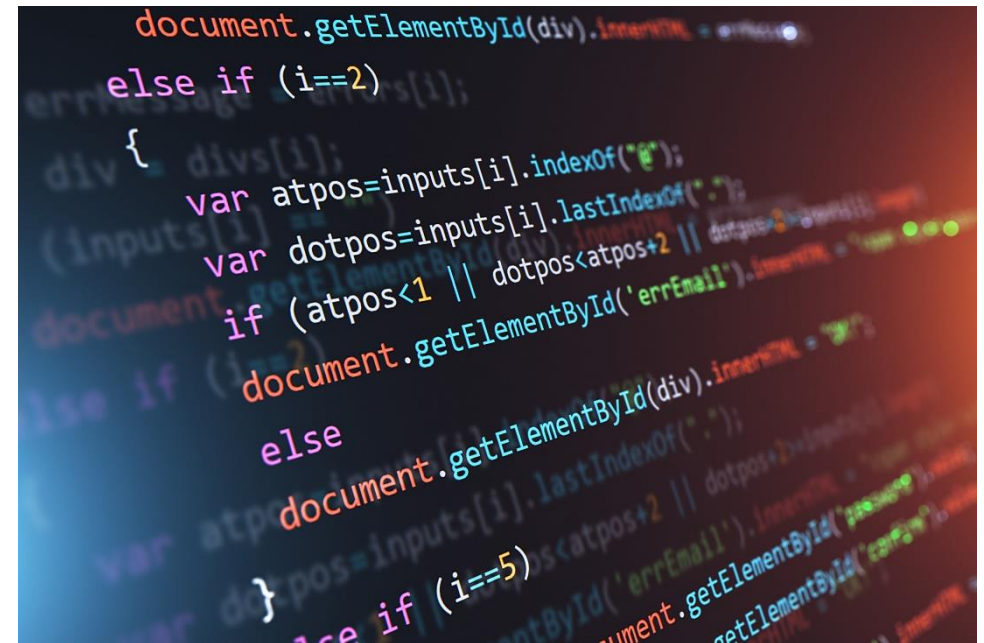




Gewerbliche  
Berufsschule  
Chur

# Informatiker/in EFZ Applikationsentwicklung

## Interner Schullehrplan





## Inhaltsverzeichnis

Information / Modulverteilung Informatiker/in EFZ Applikationsentwicklung ab 2021 .....	6
Modul 117 im 1. Semester .....	7
Modul 431 im 1. Semester .....	10
Modul 319 im 1. Semester .....	12
Modul 162 im 1. Semester .....	14
Modul 164 im 2. Semester .....	17
Modul 231 im 2. Semester .....	20
Modul 122 im 2. Semester .....	22
Modul 114 im 2. Semester .....	24
Modul 293 im 3. Semester .....	27
Modul 320 im 3. Semester .....	30
Modul 346 im 3. Semester .....	32
Modul 322 im 3. Semester .....	34
Modul 165 im 4. Semester .....	36
Modul 347 im 4. Semester .....	39
Modul 426 im 4. Semester .....	41
Modul 254 im 4. Semester .....	44
Modul 323 im 5. Semester .....	46
Modul 450 im 5. Semester .....	48
Modul 183 im 6. Semester .....	51
Modul 306 im 6. Semester .....	53
Modul 321 im 7. Semester .....	55
Modul 324 im 7. Semester .....	58
Modul 241 im 8. Semester .....	60
Modul 245 im 8. Semester .....	62

---

Mathematik 1. Semester .....	64
Mathematik 2. Semester .....	65
Mathematik 3. Semester .....	66
Mathematik 4. Semester .....	67
Englisch 1. bis 8. Semester.....	68
Beschreibung der Taxonomiestufen .....	72

## Methodisch-didaktische Überlegungen und Grundsätze

### Handlungskompetenzorientierung

Das Ziel der Berufsbildung ist die Ausbildung von arbeitsmarktfähigen Berufsfachpersonen. In der Ausbildung werden die für den Beruf relevanten beruflichen Handlungskompetenzen aufgebaut. Eine berufliche Handlungskompetenz zeigt sich in der erfolgreichen Bewältigung einer beruflichen Handlungssituation. Dazu muss eine kompetente Berufsfachperson eine situationsspezifische Kombination von relevanten Ressourcen mobilisieren. Diese sind im Bildungsplan in Form von Leistungszielen beschrieben. Welche Ressourcen in welcher Handlungssituation eingesetzt werden, müssen Lernende im Verlauf ihrer Ausbildung anhand von konkreten Situationen lernen. Aus diesem Grund reicht es nicht, nur die Ressourcen zu vermitteln, sondern es muss auch die angemessene und situationsgerechte Anwendung (Mobilisierung) der Ressourcen geübt werden. Dies geschieht vor allem in der Praxis, aber auch der Lernort Schule sollte die situationsgerechte Anwendung der Ressourcen unterstützen. Schulischer Unterricht sollte sich also möglichst auf konkrete, erlebte Situationen beziehen.

### Gestaltung des Unterrichts

Beim Handlungsorientierten Unterricht sollen die Lernenden möglichst aktiv und selbsttätig sein. Beim Aufbau des Unterrichts werden die subjektiven Interessen der Lernenden berücksichtigt und an den Erfahrungen der Lernenden angeknüpft. Um einen möglichst hohen Anteil an echter Lernzeit zu erreichen, wird die Selbsttätigkeit der Lernenden in den Vordergrund gestellt (erkunden, entdecken, erörtern und verwerfen). Die Themenstellung erfolgt situations- und problemorientiert. Um die Handlungsplanung produktbezogen zu gestalten soll im handlungsorientierten Unterricht mit möglichst konkret formulierten Arbeitsaufträgen gearbeitet werden. Das Einhalten von Spielregeln, Terminen und Qualitätsstandards soll dabei eine wichtige Rolle spielen. Der Unterricht soll so gestaltet werden, dass neben den Fach- und Methodenkompetenzen auch Sozial- und Selbstkompetenzen gefördert werden.

Wann immer möglich soll sich der Unterricht an der Lernsystematik (induktive Lernstrategie, vom Naheliegenden zum Grundlegenden) orientieren und der wissenschaftlichen Fachsystematik (deduktive Lernstrategie, vom Grundlegenden zum Besonderen, von der Theorie zur Praxis) vorgezogen werden. Die Orientierung an der Lernsystematik manifestiert sich im handlungsorientierten Unterricht unter anderem dadurch, dass Unterrichtssequenzen bei praktischen Lernsituationen beginnen und bei der dazugehörigen Theorie enden.

## Information / Modulverteilung Informatiker/in EFZ Applikationsentwicklung ab 2021

### Grundsatz:

Die Modulidentifikation ist nachstehend nicht im Detail aufgeführt. Sämtliche Informationen sind im Modulbaukasten von [ICT-Berufsbildung Schweiz](#) abrufbar. Die Modulverteilung innerhalb der Semester der Lehrjahre kann abweichen.

<b>8. Semester</b> 80 Lektionen	<b>241</b> Innovative ICT-Lösungen initialisieren	<b>245</b> Innovative ICT-Lösungen umsetzen								ENG
<b>7. Semester</b> 80 Lektionen	<b>321</b> Verteilte Systeme programmieren	<b>324</b> DevOps-Prozesse mit Tools unterstützen								ENG
<b>6. Semester</b> 80 Lektionen	<b>183</b> Applikationssicherheit implementieren	<b>306</b> Kleinprojekte im eigenen Berufsumfeld abwickeln				<b>190</b> Netzwerksicherheit implementieren				ENG
<b>5. Semester</b> 80 Lektionen	<b>323</b> Funktional Programmieren	<b>450</b> Applikationen testen				<b>185</b> Netzwerksicherheit implementieren				ENG
<b>4. Semester</b> 160 Lektionen	<b>165</b> NoSQL-Datenbanken einsetzen	<b>347</b> Dienst mit Container anwenden	<b>426</b> Software mit agilen Methoden entwickeln	<b>254</b> Geschäftsprozesse im eigenen Berufsumfeld beschreiben	<b>184</b> Netzwerksicherheit implementieren					MAT/ ENG
<b>3. Semester</b> 160 Lektionen	<b>293</b> Webauftritt erstellen und veröffentlichen	<b>320</b> Objektorientiert Programmieren	<b>346</b> Cloud Lösungen konzipieren und realisieren	<b>322</b> Benutzerschnittstellen entwerfen und implementieren	<b>188</b> Services betreiben, warten und überwachen					MAT/ ENG
<b>2. Semester</b> 160 Lektionen	<b>164</b> Datenbanken erstellen und Daten einfügen	<b>431</b> Aufträge im IT-Umfeld selbstständig durchführen	<b>122</b> Abläufe mit Scriptsprache automatisieren	<b>114</b> Codierungs-, Kompressions- und Verschlüsselungsverfahren einsetzen	<b>106</b> Datenbanken abfragen, bearbeiten und warten	<b>216</b> Internet of Everything- Endgeräte in bestehende Plattform integrieren				MAT / ENG
<b>1. Semester</b> 160 Lektionen	<b>117</b> Informatik- u. Netzinf. für kleines Unternehmen realisieren	<b>231</b> Datenschutz und Datensicherheit anwenden	<b>319</b> Applikationen entwerfen und implementieren	<b>162</b> Daten analysieren und modellieren	<b>187</b> ICT-Arbeitsplatz mit Betriebssystem in Betrieb nehmen					MAT/ ENG
Module (GBC) Fachrichtungen Applikations- und Plattformentwicklung	Module (GBC) Fachrichtung Applikationsentwicklung	Überbetriebliche Kurse (üK) Pflichtmodule Plattformentwicklung	Überbetriebliche Kurse (üK) Wahlpflichtmodule	Erweiterte Grund-Kompetenzen						

Legende: MAT = Mathematik  
ENG = Englisch

## Modul 117 im 1. Semester

Titel	Informatik- und Netzinfrastruktur für ein kleines Unternehmen realisieren		
Kompetenz	Lokales Netzwerk ohne zentrale Benutzerverwaltung installieren.		
		Taxonomie	Lektionen
	1. Mit dem Auftraggeber die Anforderungen an das zu installierende Netzwerk aufnehmen und die gewünschten Dienste/Services bestimmen (Client-Vernetzung, Internet, Cloud).	3	2
	2. Logischen und physischen Aufbau des Netzwerks gemäss betrieblichen Anforderungen und räumlichen Verhältnissen definieren und in einem Netzwerkschema und einem Verkabelungsplan abbilden.	3	12
	3. Arbeitsschritte für die Installation und Konfiguration planen, welche die Verbindung vom UGV-Anschluss zum Benutzerendgerät und die Patchung im zentralen Verteilrack berücksichtigen.	3	4
	4. Netzwerk aufbauen und die Netzwerkkomponenten gemäss Herstellerdokumentationen installieren und konfigurieren.	3	7
	5. Ressourcenverwaltung (Gruppen, Shares) und File-/ Printsharing definieren und dabei die Vorgaben hinsichtlich Datensicherheit und Verfügbarkeit berücksichtigen.	3	6
	6. Nach Vorgaben die Funktionalität, Performance, Sicherheit des Netzes, Datenzugriff der angeschlossenen Peripheriegeräte überprüfen.	3	5
	7. Netzwerk und angeschlossene Systeme testen und die Testergebnisse dokumentieren.	3	4
Kompetenzfeld	Network Management		
Objekt	Planung der Installation eines neuen lokalen Netzwerks ohne zentrale Benutzerverwaltung mit bis zu 10 Arbeitsplätzen und Internetanschluss, das Computer und Drucker in verschiedenen Räumen des gleichen Gebäudes miteinander verbindet. Installation der Netzwerkkomponenten ab der LAN-Steckdose (Computer, lokaler Drucker).		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt die verbreiteten lokalen Netzwerkdienste und kann aufzeigen, welche Anforderungen an ein Netzwerk sich daraus ergeben.	3	2
2.1	Kennt die wichtigsten Regeln für eine korrekte Netzwerkkonfiguration (IP-Adressformat, Subnetzmaske, private Adressen, Standardgateway, DNS) und kann diese anhand von Beispielen erläutern.	3	2
2.2	Kennt die wichtigsten Regeln für eine korrekte Netzwerkkonfiguration (IP-Adressformat, Subnetzmaske, Klassen, private Adressen, Standardgateway, DNS-Server) und kann diese anhand von Beispielen erläutern.	3	1
2.3	Kennt die prinzipiellen Aufgaben der Netzwerkkomponenten Switch, Accesspoint und Router und kann aufzeigen, wo und zu welchem Zweck diese in einem Netzwerk eingesetzt werden.	3	1
2.4	Kennt den Zweck und die Funktionen der Schichtenmodelle (OSI, TCP/IP-Modell) und kann die verwendeten Protokolle sowie Netzwerkkomponenten den entsprechenden Schichten zuordnen.	3	2
2.5	Kennt die verbreiteten technologischen Möglichkeiten zur Erstellung eines Internetzugangs und kann erläutern, welche Konsequenzen diese für die Nutzung des Internets und die daraus resultierenden Kosten haben.	3	2
2.6	Kennt gängige Kabeltypen, Steckertypen und Ethernet-Varianten (z.B. Twisted Pair, UTP, STP, Glasfaser, RJ45, etc.) und kann aufzeigen, bei welchen Anforderungen hinsichtlich Leistung und bei welchen räumlichen Gegebenheiten diese zum Einsatz kommen.	3	1
2.7	Kennt die Vorgehensweise, ein Netzwerk sowohl in einer logischen wie einer physischen Darstellung abzubilden.	3	1
2.8	Kennt die gängigen WLAN Standards.	3	1
2.9	Kennt die erforderlichen Arbeitsschritte und Komponenten, um eine einfache WLAN-Vernetzung einzurichten.	3	1
3.1	Kennt relevante bauliche Gegebenheiten und Installationsmöglichkeiten hinsichtlich der Netzwerk-Verkabelung und kann deren Auswirkungen auf Installationsaufwand, Zugänglichkeit für den Unterhalt und Kosten aufzeigen.	3	4
4.1	Kennt die notwendigen Einstellungen der Netzwerkkonfiguration der Netzwerkkomponenten (z.B. Computer, Router, Accesspoint) und kann aufzeigen, welchen Beitrag diese zur Sicherstellung der Kommunikation im Netzwerk leisten.	3	4





Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
4.2	Kennt die wichtigsten Informationen in der Dokumentation eines einfachen Netzwerks und kann erläutern, wie diese für die Wartung und den Betrieb benötigt werden.	3	3
5.1	Kennt die prinzipiellen Vorkehrungen, die Netzwerkbetriebssysteme für die Ressourcenzuteilung bieten (Lese-, Schreibrecht, Benutzer, Benutzergruppen, Shares) und kann aufzeigen, wie diese die Sicherheit von Daten gewährleisten.	3	2
5.2	Kennt die prinzipiellen Vorkehrungen, die bei Cloudspeicherung sicherzustellen sind und kann aufzeigen, wie diese die Sicherheit von Daten gewährleisten.	3	2
5.3	Kennt Möglichkeiten, die Vergabe von Rechten zu dokumentieren (z.B. Matrix der Beziehungen zwischen Benutzergruppen und Shares) und kann aufzeigen, wie damit eine korrekte Vergabe der Rechte erleichtert wird.	3	2
6.1	Kennt die gängigen Vorgehensweisen und Methoden, einen Test mit den dazugehörigen Testszenarien durchzuführen.	3	1
6.2	Kennt die Vorgehensweise, einen geplanten funktionalen Test durchzuführen.	3	2
6.3	Kennt die Symptome der wichtigsten Fehler in einem Netzwerk und kann mögliche Ursachen (Konfigurationsfehler, Fehler bei der Verkabelung etc.) dafür beschreiben.	3	2
7.1	Kennt die Möglichkeiten den Datenschutz und -sicherheit zu testen und die Ergebnisse zu dokumentieren.	3	2
7.2	Kennt die Möglichkeiten das Netzwerk und all seine Elemente zu testen und die Ergebnisse zu dokumentieren.	3	2

## Modul 431 im 1. Semester

Titel		Aufträge im eigenen Berufsumfeld selbstständig durchführen			
Kompetenz	Führt Aufträge aus dem eigenen Berufsumfeld gemäss Vorgaben des Auftraggebers selbständig und mit Hilfe geeigneter Techniken, Methoden und Hilfsmittel durch.				
				Taxonomie	Lektionen
Handlungsziele	1.	Nimmt einen Auftrag entgegen, analysiert ihn und klärt Unklarheiten mit dem Auftraggeber.		4	6
	2.	Plant einen Auftrag gemäss dem Prinzip der vollständigen Handlung.		4	8
	3.	Führt einen Auftrag unter Verwendung geeigneter Techniken und Methoden durch.		3	16
	4.	Dokumentiert das Ergebnis eines Auftrags, präsentiert dieses dem Auftraggeber und hinterfragt das Ergebnis kritisch.		3	10
Kompetenzfeld	Project Management				
Objekt	Aufträge im eigenen Berufsumfeld mit definierten Zielen und Ergebnissen				
Anzahl Lektionen	40				
Anerkennung	Eidg. Fähigkeitszeugnis				

Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt die grundsätzlichen Elemente einer Organisation (Aufgabe, Aufgabenträger, Sachmittel, Information) und deren Beitrag zur Bestimmung einer Aufgabe.	4	2
1.2	Kennt den generellen Phasenablauf eines Projektes (z.B. Initialisierung, Konzept, Realisierung, Einführung) als Umfeld einer Auftragsbearbeitung	4	2
1.3	Kennt die typischen Rollen von Auftraggeber und Auftragnehmer sowie deren Aufgaben, Kompetenzen und Verantwortungen.	4	2
2.1	Kennt ein Modell der vollständigen Handlung.	4	2
2.2	Kennt eine Technik zur Steuerung des Projektes, die dazu beiträgt, die Einhaltung der Ziele und der Vorgaben eines Auftrages zu unterstützen.	4	3
2.3	Kennt den Mehrwert von Checklisten, die zur Bewältigung eines Auftrags eingesetzt werden.	4	3
3.1	Kennt verschiedene Problemlösungsmethoden und deren Einsatzbereiche (z.B. 5W Methode, Ishikawa-Diagramm).	3	4
3.2	Kennt verschiedene Kreativitätstechniken und deren Einsatzbereiche.	3	6
3.3	Kennt verschiedene Lerntechniken, um sich das nötige Wissen für eine Auftragsbearbeitung anzueignen.	3	6
4.1	Kennt die Bedeutung einer Dokumentation zur Sicherstellung und Nachvollziehbarkeit von Arbeitsergebnissen	3	3
4.2	Kennt die wichtigsten inhaltlichen und formalen Regeln, die bei der Dokumentation von Arbeitsergebnissen einzuhalten sind.	3	3
4.3	Kennt verschiedene Präsentationstechniken und deren Einsatzbereiche.	3	2
4.4	Kennt Möglichkeiten, die Auftragsabwicklung zu reflektieren (z.B. Sozialverhalten, Kommunikation, Arbeitstechnik, Lerntechnik).	3	2

## Modul 319 im 1. Semester

Titel	Applikationen entwerfen und implementieren		
Kompetenz	Die Lernenden kennen die Grundlagen des Programmierens, deren Herkunft, ihrer Voraussetzungen und sind in der Lage im Beruflichen Umfeld Probleme zu verstehen und Lösungen dafür zu entwickeln.		
		Taxonomie	Lektionen
	1. Erfasst Problemstellungen, entwickelt strukturiert Lösungsansätze und übersetzt sie für die Stakeholder.	3	6
	2. Erstellt eine geeignete visuelle Darstellung für die Programmierung von Anforderungen.	3	5
	3. Leitet aufgrund der Vorgaben die erforderlichen Daten (Eingabe, Verarbeitung, Ausgabe und ihre Datentypen) ab.	3	6
	4. Implementiert die Applikation mit Hilfe von Kontrollstrukturen und selbst erstellten Funktionen.	3	15
	5. Hält vorgegebene Konventionen ein, kommentiert den Code und achtet dabei auf die Wartbarkeit.	3	2
	6. Interpretiert Mängel (Fehler) in der Software und korrigiert diese.	3	6
Kompetenzfeld	Application Engineering		
Objekt	Problemstellungen, welche mit einer Programmiersprache umgesetzt werden können.		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt den Unterschied und den Zweck von Problemraum zu Lösungsraum.	3	2
1.2	Kennt Instrumente, um Anforderungen Lösungsfrei zu identifizieren.	3	2
1.3	Kennt die Gründe, um den Problemraum und Lösungsraum voneinander zu trennen (z.B. Innovationsgewinn, Stressreduktion).	3	2
2.1	Kennt eine Notation, um den Ablauf einer Programmlogik abzubilden (z. B. PAP, Struktogramm, Activity).	3	2
2.2	Kennt den Zweck von Modellen und Diagrammen als Besprechungs- und Umsetzungsgrundlage.	3	3
3.1	Kennt die Unterscheidung der Datentypen (z. B. primitive, komplexe).	3	2
3.2	Kennt die relevanten Datentypen (z. B. integer, boolean, array mit einer Dimension).	3	2
3.3	Kennt die Funktionsweise einer Software nach dem EVA Prinzip.	3	2
4.1	Kennt den grundlegenden Aufbau einer Applikation resp. Programmes.	3	3
4.2	Kennt die Verwendung von Kontrollstrukturen wie Iterationen (vor-/nachgeprüft), Selektion.	3	2
4.3	Kennt Sinn und Zweck von Variablen und Konstanten sowie deren Unterschied.	3	1
4.4	Kennt den Gültigkeitsbereich von Anweisungen und deren Zweck.	3	2
4.5	Kennt die Möglichkeiten von Editoren sowie einer Entwicklungsumgebung, sowie den Umgang mit Fehler- und Warnmeldungen.	3	2
4.6	Kennt den Aufbau von Funktionen (Name, Parameter, Datentypen, Rückgabetyt) und wie diese in einem Programm korrekt eingesetzt und aufgerufen werden.	3	3
4.7	Kennt den Mehrwert der Auslagerung von Teilen von Code in Funktionen.	3	2
5.1	Kennt die Bedeutung von Konventionen (z.B. Firmenkonventionen, Sprachkonventionen).	3	1
5.2	Kennt die Bedeutung von Kommentaren und deren Beitrag zu besser wartbarer Software.	3	1
6.1	Kennt den Zweck, die Funktionen und die Bedienung eines Debugger.	3	2
6.2	Kennt Sinn und Zweck des Testens, sowie die Absicht von einzelnen Testfällen.	3	2
6.3	Kennt ein Verfahren für den statischen Test (z.B. Codereview) von Programmcode und kann die Bedeutung für die Qualität der Software aufzeigen.	3	2

## Modul 162 im 1. Semester

Titel		Daten analysieren und modellieren			
Kompetenz	Analysiert Informationsbestände aus verschiedenen Quellen, charakterisiert Daten und ordnet sie ein. Erstellt durch die in der Analyse gewonnenen Anhaltspunkte ein konzeptionelles Datenmodell und überführt dieses in ein logisches, relationales Datenmodell.				
				Taxonomie	Lektionen
	1.	Sichtet Daten aus verschiedenen strukturierten und unstrukturierten Informationsbeständen und ordnet sie ein.		3	4
	2.	Charakterisiert einen Informationsbestand für die Verarbeitung hinsichtlich Qualität, z.B. Vollständigkeit, Eindeutigkeit, Widersprüchlichkeit, Redundanz und bereinigt die Daten falls nötig.		3	6
	3.	Wählt Informationen aus, welche sich für eine Auswertung eignen und berücksichtigt Aspekte bezüglich allfälliger Schutzbedürfnisse (z.B. Personendaten, Geheimhaltung).		3	6
	4.	Erstellt ein konzeptionelles Datenmodell für die bereinigten Daten und definiert Entitätstypen, Attribute und Beziehungen.		3	8
	5.	Überführt das konzeptionelle Datenmodell in ein logisches, relationales Datenmodell durch Ergänzen von Identifikations- und Fremdschlüsseln, Datentypen und allfälliger Zwischentabellen.		3	8
	6.	Normalisiert das logische, relationale Datenmodell.		3	4
	7.	Bildet das konzeptionelle und logische, relationale Datenmodell in einer geeigneten Darstellung ab.		3	4
Kompetenzfeld	Data Management				
Objekt	Datenbestände (Texte, Listen, einzelne Tabellen, Messreihen, Datenbanken mit bis zu fünf Tabellen, komplexe und rekursive Beziehungstypen, z.T. schützenswerte Informationen).				
Anzahl Lektionen	40				
Anerkennung	Eidg. Fähigkeitszeugnis				



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt verschiedene Formen von Daten (numerische: Messdaten/Klassifikationen; verbale: Texte in schriftlicher oder gesprochener Form; nonverbale: Musik, Fotos, Videos, usw.) und wie sie auf einem Datenträger gespeichert werden. Kennt deren Auswertbarkeit auf einem Computersystem.	3	2
1.2	Kennt Strukturmerkmale verschiedener Informationsquellen (Textdokumente, Datenblätter, Geschäftsgrafiken, Datenbanken, usw.) und die Konsequenzen, welche sich daraus für die elektronische Auswertung dieser Datenquellen ergeben.	3	2
2.1	Kennt Merkmale eines Datenbestandes bezüglich Eindeutigkeit und Vollständigkeit (unterschiedliche Angaben zur gleichen Information, unterschiedlicher Informationsgehalt).	3	1
2.2	Kennt Merkmale eines Datenbestandes bezüglich Eindeutigkeit und Vollständigkeit (unterschiedliche Angaben zur gleichen Information, unterschiedlicher Informationsgehalt).	3	1
2.3	Kennt die verschiedenen Skalentypen (nominal, ordinal und metrisch) und die sich daraus ergebenden Konsequenzen für die Auswertungsmöglichkeiten.	3	2
3.1	Kennt verbreitete Diagrammtypen für die Darstellung von Daten in Geschäftsgrafiken (Kreisdiagramm, Balkendiagramm, Liniendiagramm, Netzdiagramm, usw.).	3	2
3.2	Kennt die Notwendigkeit, Auswertungsergebnisse auf Relevanz zu beurteilen und dass Tendenzen in diesen je nach Perspektive relevant, aber auch nicht relevant sein können.	3	2
3.3	Kennt wichtige statistische Kenngrößen (Minimum, Maximum, Mittelwert, Median, Durchschnitt, Anzahl Beobachtungen) und die Voraussetzungen (Skalentypen, Anzahl Beobachtungen, Qualität der Daten, usw.) für eine zuverlässige Aussage.	3	2
3.4	Kennt Grundprinzipien von Geheimhaltung und Datenschutz bei der Erfassung, Auswertung und Darstellung von schützenswerten Datenbeständen.	3	2
4.1	Kennt die methodische Vorgehensweise bei der Entwicklung eines konzeptionellen Datenmodells (Entity Relationship Model, ERM), ausgehend von Datenstrukturen bzw. Anforderungen des Kunden (spezialisieren, generalisieren, Entitätstypen identifizieren und definieren, Attribute definieren, Beziehungen beschreiben).	3	3
4.2	Kennt Elemente in einem konzeptionellen Datenmodell (Entitätstyp, Attribut, Assoziation, Kardinalität), deren Eigenschaften und deren Darstellung.	3	3
4.3	Kennt die verschiedenen Kardinalitäten von Assoziationen (1:1, 1:m, n:m) und kennt das Vorgehen, um den Beziehungstyp zwischen Datenstrukturen zu bestimmen.	3	2



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
5.1	Kennt die Unterschiede zwischen dem konzeptionellen und dem logischen, relationalen Datenmodell (Relational Model, RM) und das Vorgehen zur Überführung vom konzeptionellen zum logischen, relationalen Datenmodell.	3	2
5.2	Kennt zu erfüllende Merkmale von Identifikationsschlüsseln und die daraus entstehenden Verarbeitungsmöglichkeiten als Primär- und Fremdschlüssel.	3	2
5.3	Kennt die Notwendigkeit von Zwischentabellen zur Abbildung von n:m-Kardinalitäten zwischen Entitätstypen.	3	2
5.4	Kennt erweiterte Konstruktionselemente (z.B. Rekursive Assoziation) des logischen, relationalen Datenmodells.	3	2
6.1	Kennt die Schritte der Normalisierung von der 0. bis zur 3. Normalform.	3	2
6.2	Kennt Gründe, weshalb eine vollständige Normalisierung bis zur 3. Normalform zugunsten der besseren Performanz nicht immer angestrebt wird.	3	2
7.1	Kennt Notationen zur Darstellung von konzeptionellen und logischen, relationalen Datenmodellen (IEM- bzw. Crow foot-, Chen-, Modified Chen-, Zehnder-, UML-Notation usw.).	3	2
7.2	Kennt Tools, um konzeptionelle und logische, relationale Datenmodelle zu erstellen.	3	2



## Modul 164 im 2. Semester

Titel		Datenbanken erstellen und Daten einfügen	
Kompetenz	Implementiert ein logisches, relationales Datenmodell in einem Datenbankmanagementsystem. Fügt Daten in die Datenbank ein, prüft die eingefügten Daten und korrigiert allfällige Fehler.		
		Taxonomie	Lektionen
	1. Interpretiert die Darstellung eines logischen, relationalen Datenmodells korrekt.	3	2
	2. Implementiert ein logisches, relationales Datenmodell in einem relationalen Datenbankmanagementsystem.	3	12
	3. Stellt die referenzielle Integrität im implementierten Datenbankschema durch Integritätsbedingungen (Constraints) sicher.	3	6
	4. Fügt Daten mittels Datenbearbeitungssprache (DML) in die Datenbank ein.	3	6
	5. Importiert Daten in die Datenbank (z.B. aus Dateien oder anderen Tabellen).	3	4
	6. Prüft die Vollständigkeit und Korrektheit der eingefügten Daten durch einfache Abfragen.	3	8
	7. Korrigiert fehlerhafte und unvollständige Daten.	3	2
Kompetenzfeld	Data Management		
Objekt	Datenbanken mit bis zu zehn Tabellen (einfache, komplexe und rekursive Beziehungstypen) und schützenswerten Daten (z.B. Kunden- oder Patientenverwaltung).		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt Vorgehensweise, um ein logisches, relationales Datenmodell systematisch zu interpretieren.	3	2
2.1	Kennt die Elemente (Datenbank, Zeichensatz, Schema, Tablespace, Tabelle, Partition, Feld, usw.) eines Datenbankmanagementsystems (DBMS) zur Umsetzung eines Datenmodells in ein physisches Datenbankschema.	3	4
2.2	Kennt die Befehle einer Datendefinitionssprache (DDL) zur Implementierung des logischen, relationalen Datenmodells in einem physischen Datenbankschema.	3	4
2.3	Kennt gebräuchliche Datentypen (Zeichenfolgen fix und variabel, Ganzzahlen, Fließkommazahlen, Datum und Zeit, Binärtypen) und deren Grenzen und Verwendung passend zum logischen Datentyp.	3	2
2.4	Kennt Möglichkeiten zur Vergabe von Standardwerten für Attribute und zur automatischen Verwaltung des Primärschlüssels.	3	2
3.1	Kennt Elemente und Befehle einer Datendefinitionssprache (DDL) zur Implementierung von Integritätsbedingungen (Constraints), d.h. zur Sicherstellung der referenziellen Integrität.	3	2
3.2	Kennt weitere Constraints auf Tabellen- und Attributebene (keine leeren Werte, Einschränkung des Wertebereichs, Standardwerte, usw.) und kennt die möglichen Regeln beim Ändern und Löschen eines Datensatzes.	3	2
3.3	Kennt Befehle zur zeitweisen Aussetzung der Integritätsbedingungen, z.B. zur Ermöglichung oder Beschleunigung von Imports.	3	2
4.1	Kennt Befehle einer Datenbearbeitungssprache (DML) zum Hinzufügen einzelner oder mehrerer Datensätze in die Datenbank (über die Konsole, aus bestehenden Tabellen, über DBMS-Frontend Tools).	3	6
5.1	Kennt Varianten und Befehle einer Datenbearbeitungssprache (DML) zum Hinzufügen von Datensätzen über einen Massenimport (Bulk Load) aus externen Quellen (CSV, XML, JSON, SQL-Insert, usw.).	3	2
5.2	Kennt Befehle einer Datenbearbeitungssprache (DML) zur Manipulation von Datensätzen während des Massenimports.	3	2
6.1	Kennt Testkriterien zur Prüfung der importierten Daten auf Vollständigkeit und Datenintegrität.	3	2



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
6.2	Kennt Befehle einer Datenbearbeitungssprache (DML) zur Abfrage von Datenbeständen und nutzt einfache Filterfunktionen.	3	4
6.3	Kennt Befehle einer Datenbearbeitungssprache (DML) zur Erstellung von Checksummen zur abschliessenden Prüfung der Datenbestände.	3	2
7.1	Kennt Befehle einer Datenbearbeitungssprache (DML) zur Anpassung oder Entfernung einzelner oder mehrerer Datensätze.	3	2

## Modul 231 im 2. Semester

Titel		Datenschutz und Datensicherheit anwenden	
Kompetenz	Setzt Datenschutz und Datensicherheit bei Informatiksystemen ein. Überprüft vorhandene Systeme auf Einhaltung von Richtlinien.		
		Taxonomie	Lektionen
	1. Kategorisiert Daten aufgrund ihres Schutzbedarfs.	3	10
	2. Überprüft und verbessert gegebenenfalls die Datensicherheit der eigenen Infrastruktur.	3	16
	3. Setzt verschiedene Möglichkeiten der Datenspeicherung ein.	3	6
	4. Überprüft eingesetzte Anwendungen auf Einhaltung der Datenschutzgesetze.	3	2
	5. Zeigt Konsequenzen von Fehlern im Datenschutz und bei der Datensicherheit auf.	3	4
	6. Wählt Software für die Einhaltung von Datenschutz und Datensicherheit aufgrund der Lizenzmodelle aus.	3	2
Kompetenzfeld	Security/Risk Management		
Objekt	Daten und Informationen aus dem beruflichen Umfeld		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt verschiedene Kategorien der Schutzwürdigkeit von Daten und deren Kriterien.	3	4
1.2	Kennt den Unterschied von Datenschutz und Datensicherheit.	3	2
1.3	Kennt verschiedene Rechtsräume (Schweiz, EU).	3	2
1.4	Kennt für den jeweiligen Rechtsraum die juristischen Werke (z. B. DSG, DSGVO).	3	2
2.1	Kennt Möglichkeiten zur Verschlüsselung von Daten auf dem eigenen Rechner (z.B. Datenträger-verschlüsselung).	3	4
2.2	Kennt Verfahren zur Erstellung und Wiederherstellung von Backups.	3	4
2.3	Kennt Techniken des Zugriffsschutzes, Passwortmanager und Prinzipien der Passwortverwaltung.	3	4
2.4	Kennt den Unterschied von Authentifizierung und Autorisierung.	3	4
3.1	Kennt Verfahren zur Speicherung von Daten und bewusst redundanter Datenhaltung (z. B. lokal, Server, Cloud).	3	4
3.2	Kennt verschiedene Gefahren, denen Daten ausgesetzt sind (z.B. Diebstahl, Ransomware, Integritätsverletzung).	3	2
4.1	Kennt wesentliche Unterschiede in den Datenschutzgesetzen der verschiedenen Rechtsräume.	3	2
5.1	Kennt die Problematik von Datenlöschungen über alle Archive und Backups.	3	2
5.2	Kennt wesentliche juristische Voraussetzungen und Eigenheiten von Websites (z. B. Impressum, Disclaimer, AGBs).	3	2
6.1	Kennt verschiedene Lizenzmodelle (z. B. für Software, Texte, Bilder).	3	2

## Modul 122 im 2. Semester

Titel	Abläufe mit einer Scriptsprache automatisieren		
Kompetenz	Abläufe mit Scripts in der Systemadministration automatisieren.		
		Taxonomie	Lektionen
	1. Zu automatisierende Funktion oder zu automatisierenden Ablauf mit den dazugehörigen Benutzerinteraktionen als Ablaufstruktur (z.B. Programmablaufplan) grafisch darstellen.	3	10
	2. Ablaufstruktur mit Hilfe einer Scriptsprache umsetzen.	3	14
	3. Script in eine Systemumgebung integrieren.	3	8
	4. Script auf eine vollständige und korrekte Ausführung der erforderlichen Funktionalität bzw. des Ablaufs testen.	3	4
	5. Dokumentation für den Einsatz des Scripts erstellen.	3	4
Kompetenzfeld	Service Management		
Objekt	Funktionen und Abläufe mit Kontrollstrukturen (Sequenz, Selektion und Iteration) mit dazugehörigen Benutzerinteraktionen.		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt Beispiele von Automatisierungsaufgaben.	3	1
1.2	Kennt Kriterien, die für die Beurteilung des Automatisierungspotentials von Bedeutung sind.	3	1
1.3	Kennt grundlegende Kontrollstrukturen und deren Einsatz bei der Ablaufautomatisierung.	3	2
1.4	Kennt das Vorgehen bei der Situationsanalyse und weiss, welchen Beitrag diese zur Definition eines klaren Auftrags liefert.	3	2
1.5	Kennt das Vorgehen zur grafischen Darstellung von Abläufen.	3	2
1.6	Kennt das Vorgehen zur Realisierung von Scripts in der Systemadministration.	3	2
2.1	Kennt grundlegende Funktionalitäten der eingesetzten Scriptsprache.	4	14
3.1	Kennt gängige Sicherheitsmassnahmen für den Einsatz von Scripts.	4	4
3.2	Kennt Integrationsmöglichkeiten von Scripts im eingesetzten Betriebssystem.	4	4
4.1	Kennt ein Testverfahren für Scripts.	4	2
4.2	Kennt das Testvorgehen zur Integration von Scripts.	4	2
5.1	Kennt die Elemente einer Dokumentation für die involvierten Rollen (z.B. System, Administrator, Entwickler)	3	2
5.2	Kennt die Bedeutung der Dokumentation in Bezug auf Qualitätssicherung und Wartbarkeit.	3	2

## Modul 114 im 2. Semester

Titel		Codierungs-, Kompressions- und Verschlüsselungsverfahren einsetzen	
Kompetenz	Codierungs-, Kompressions- und Verschlüsselungsverfahren im täglichen Berufsalltag korrekt einsetzen.		
		Taxonomie	Lektionen
	1. Codierungen von Daten situationsbezogen auswählen und einsetzen. Aufzeigen, welche Auswirkung die Codierung auf die Darstellung von Daten hat.	3	12
	2. Kompressionsverfahren gemäss Vorgaben für die Aufbewahrung, Wiederherstellung und Übertragung von Daten auswählen und einsetzen.	3	6
	3. Verschlüsselungsverfahren zur Sicherung von Daten gemäss Vorgaben gegen unbefugten Zugriff auf Datenspeicher und Übertragungswegen auswählen und einsetzen.	3	10
	4. Gesicherte Übertragungsverfahren für Dateien mit asymmetrischen und symmetrischen Verschlüsselungsverfahren nutzen. Dabei Aspekte wie Public/Private Key, Zertifikate, Protokolle und Standards berücksichtigen.	3	8
	5. Verschiedene Verschlüsselungstechnologien hinsichtlich Aktualität, Verbreitung und Sicherheit bewerten. Schwachstellen erkennen und Vorschläge für alternative Technologien machen.	3	4
Kompetenzfeld	Security/Risk Management		
Objekt	Zu speichernde und zu übertragende Daten in einem Unternehmen.		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		





Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt die wichtigsten Typen von Binärcodes (z.B. ASCII, ANSI-, BCD-, EAN-, 1-aus -n-, UTF, Uni-Code) und kann anhand ihrer Merkmale (Zeichenvorrat, Redundanz) aufzeigen, wie sich diese hinsichtlich der Bewertbarkeit, Fehlererkennbarkeit und Rechenfähigkeit unterscheiden.	3	4
1.2	Kennt die wichtigsten Eigenschaften von Bildern (z. B S/W Strichzeichnung, Farbfoto, bewegte/nicht bewegte Bilder, vektorisiert/pixelorientiert usw.) und kann erläutern, wie damit die Bildqualität (z. B Auflösung, Farbtiefe), der Bildaufbau und der Speicherbedarf beeinflusst werden kann, wie auch Sicherheitslücken bezüglich Metadaten aufzeigen.	3	2
1.3	Kennt binäre, oktale und hexadezimale Zahlensysteme, logische Operationen (OR, AND, NOT) und weiss wie diese in der IT eingesetzt inkl. deren Umrechnung in andere Zahlen-systeme werden (z. B Unix-Dateirechte, IP-Adressen, Farben-RGB usw.).	3	3
1.4	Kennt Verfahren zur binären Kodierung von Zahlen (z. B negative Zahlen/Zweierkomplement, Gleitkommazahlen, Exzess).	3	3
2.1	Kennt Merkmale (z. B Kompressionsrate, Qualitätsverlust) für gängige verlustlose und verlustbehaftet Kompressionsverfahren und an welchen Stellen, welches sinnvoll eingesetzt wird.	3	3
2.2	Kennt ausgewählte Normen und Standards (z. B. JPEG, PNG, MPEG, H261/263, Huffmann-Verfahren) und kennt typische Anwendungsbereiche wo diese eingesetzt werden, wie auch die Berücksichtigung von mehrfach komprimierten Dateien ( z. B JPEG in einem *.Zip) und deren nachteilige Folgen.	3	3
3.1	Kennt das grundsätzliche Prinzip der Verschlüsselung von Informationen (z. B Kryptografie/Steganografie) und kann anhand eines einfachen Verschlüsselungskonzepts aufzeigen, wie damit Informationen chiffriert und dechiffriert werden können.	4	6
3.2	Kennt mögliche aktuelle oder zukünftige Applikationen, welche für die Verschlüsselung von Daten (z. B. PGP, Keypass) eingesetzt werden können und an welchen Stellen diese Applikationen sinnvoll sind.	4	2
3.3	Kennt die Konfiguration der Verschlüsselungsapplikationen und kann diese gemäss Firmenvorgabe oder Anleitung korrekt in Betrieb nehmen.	4	2
4.1	Kennt die prinzipiellen Unterschiede zwischen einer symmetrischen und asymmetrischen Verschlüsselung (z. B Passwörter, private und öffentliche Schlüssel) und kann erläutern, wie sich diese auf den Grad der Datensicherheit auswirken.	4	4



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
4.2	Kennt das Prinzip elektronischer Signatursysteme und kann anhand von Beispielen aufzeigen, wie damit die Sicherheit (z. B Authentizität, Integrität) der Übermittlung gewährleistet werden kann, wie auch an welchen Stellen heute und in Zukunft diese eingesetzt werden können.	3	2
4.3	Kennt den Zweck digitaler Zertifikate und kann an Beispielen erläutern, wie damit das Vertrauen zwischen Anbieter und Bezüger (einer Leistung, eines Produktes) sichergestellt werden kann.	3	2
5.1	Kennt mögliche IT bezogene Stellen, wie API-Schnittstellen, Versenden von E-Mails, wie diese bei der Übertragung von Daten verschlüsselt werden sollten und welche minimale Datenverschlüsselung bei der Übertragung gemäss Gesetz verwendet werden muss (z. B Bit-Länge der Verschlüsselung).	3	2
5.2	Kennt mögliche Schwachstellen der Verschlüsselungsverfahren bei Datenverschlüsselung oder Datenübertragung und kann diese aufzeigen.	3	2

## Modul 293 im 3. Semester

Titel	Webauftritt erstellen und veröffentlichen		
Kompetenz	Erstellt einen Webauftritt mit modernen Webtechnologien und -sprachen nach Vorgaben und veröffentlicht ihn.		
		Taxonomie	Lektionen
	1. Überprüft einen Gestaltungsentwurf (Mockup, klickbarer Prototyp) einer Webseite auf Machbarkeit.	3	6
	2. Entwickelt eine lauffähige Webseite als «Proof of concept» für eine erste Überprüfung.	3	14
	3. Entwickelt das «Proof of concept» weiter zu einem finalisierten Webauftritt und überprüft ihn auf Erfüllung der Vorgaben und Richtlinien aus der Aufgabenstellung.	3	14
	4. Veröffentlicht einen Webauftritt auf einem Webserver.	3	4
Kompetenzfeld	Web Engineering		
Objekt	Webauftritt mit verschiedenen Elementen (bspw. Text, Grafik, Animation, Links) und ohne Server-/Clientseitige Programmierung. Zum Beispiel: Webauftritt für einen kleinen Verein.		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt wichtige Prinzipien (z.B. Request/Response, Stateless, Asynchron) für die Webentwicklung.	3	2
1.2	Kennt relevante Aspekte, unter welchen die Machbarkeit überprüft werden kann.	3	2
1.3	Kennt die grundlegenden Unterschiede der verschiedenen Browser (z.B. Firefox, Chrome, Safari) hinsichtlich Kompatibilität, Funktionalität und Darstellung.	3	2
2.1	Kennt geeignete Entwicklungs- und Laufzeitumgebungen (z.B. IntelliJ, Visual Studio Code) und kennt die Vorgehensweise, sie nach Vorgaben einzurichten.	3	3
2.2	Kennt Grundelemente einer Webseitensprache für Semantik (z.B. HTML)	3	2
2.3	Kennt die Möglichkeiten (z.B. CSS, SCSS), eine Webseite zu gestalten (Layout und Design).	3	3
2.4	Kennt Grundelemente einer Webseitensprache für Interaktion und Navigation und Navigation (z.B. JavaScript).	3	3
2.5	Kennt die Unterschiede, welche Sprachen und Technologien von einem Browser direkt unterstützt werden (z.B. CSS, JS), welche umgewandelt werden müssen (z.B. SCSS, Typescript) und kennt Werkzeuge (z.B. Babel, Typescript), mit welchen diese umgewandelt werden.	3	3
3.1	Kennt Methoden, wie lesbarer und wartbarer Code geschrieben wird.	3	3
3.2	Kennt die empfohlenen Vorgehensweisen (Best Practices), wie eine Webseite aufgebaut und programmiert wird.	3	3
3.3	Kennt Techniken (Responsive, Cross-Browser-Kompatibilität), wie eine Webseite auf verschiedenen Endgeräten (z.B. Desktop, Mobile) optimal angezeigt wird.	3	3
3.4	Kennt eine Methode (z.B. Testen, Validierung) für die Erfolgskontrolle sowie deren Einsatz für die Bestimmung der Zielerfüllung.	3	3
3.5	Kennt Coderichtlinien und Methoden, wie diese überprüft werden können (z.B. W3C-Validatoren, Einsatz eines Tools, IDE, Linter).	3	2
4.1	Kennt verschiedene Möglichkeiten, wie eine Webseite von einer lokalen Umgebung auf einen Server gebracht und veröffentlicht wird.	3	2
4.2	Kennt die rechtlichen und sicherheitstechnischen Aspekte, die bei einem Webauftritt berücksichtigt und eingehalten werden müssen (z.B. Urheberrecht, Persönlichkeitsschutz).	3	2



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
4.3	Kennt die rechtlichen und sicherheitstechnischen Abhängigkeiten, die im Zusammenhang mit dem geographischen Standort des Webservers stehen (z.B. europäischer Datenschutz, amerikanischer Datenschutz, Schweizer Datenschutz).	3	2

## Modul 320 im 3. Semester

Titel	Objektorientiert Programmieren		
Kompetenz	Applikationen und Schnittstellen objektorientiert modellieren, implementieren, testen und dokumentieren.		
		Taxonomie	Lektionen
	1. Analysiert Anwendungsprobleme zur Erstellung von objektorientierten Programmen.	3	6
	2. Modelliert und dokumentiert objektorientierte Programme	3	12
	3. Implementiert objektorientiertes Design.	3	14
	4. Überprüft die Implementierung auf Korrektheit und Qualität	3	8
Kompetenzfeld	Application Engineering		
Objekt	Anwendung mit 3 bis 5 fachlichen Klassen.		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		

Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt den objektorientierten Ansatz mit den Grundkonzepten Kapselung, Vererbung und Polymorphie.	3	2
1.2	Kennt Vorgehensprinzipien zur Klassenfindung (z.B. Spezialisierung und Generalisierung, Domänenmodell und –Events, BoundedContext, DRY, SRP)	3	2
1.3	Kennt Abstraktionskonzepte wie Assoziationen, Klassen, Attribute und Methoden, Schnittstellen und abstrakte Klassen und Datentypen (ADT, Collections, Generics).	3	2
2.1	Kennt Beschreibungen (z.B. CRC-Card) und Diagrammnotationen (z.B. UML, TAM, 4+1) für statische und dynamische Aspekte von objektorientiertem Design.	3	6
2.1	Kennt das Dokumentieren mit einer Dokumentationsinfrastruktur (z.B. Javadoc).	3	6
3.1	Kennt den Unterschied zwischen Klassen und Objekten.	3	4
3.2	Kennt eine objektorientierte Sprache und deren Elemente und kann damit das Design umsetzen.	3	4
3.3	Kennt das Konzept der dynamischen Bindung.	3	3
3.4	Kennt das Konzept der Inversion of Control (DependencyInjection)	3	3
4.1	Kennt Verfahren zum Erstellen von Testfällen.	3	4
4.2	Kennt Testmethoden zur Überprüfung der Komponenten einer Anwendung (automatisches Unit-Testing).	3	4

## Modul 346 im 3. Semester

Titel	Cloud Lösungen konzipieren und realisieren		
Kompetenz	Beurteilt die Eignung von Cloud Services hinsichtlich der betrieblichen Anforderung. Entwickelt ein technisches Konzept und realisiert die gewählte Lösung.		
	<ol style="list-style-type: none"> <li data-bbox="488 564 1733 628">1. Beurteilt die Eignung von On-Premise und Cloudlösungen abgestimmt auf die Zielsetzungen des Unternehmens und leitet draus eine Empfehlung für die Umsetzung ab.</li> <li data-bbox="488 633 1733 697">2. Spezifiziert die Kosten einschliesslich Betriebsaufwand der vorgeschlagenen Lösung und bestimmt die zweckmässige Cloud Adoption.</li> <li data-bbox="488 702 1733 793">3. Entwickelt unter Berücksichtigung der technischen Rahmenbedingungen und Anforderungen des entsprechenden Anwendungsbereichs ein technisches Konzept für die Integration der ausgewählten Cloudlösung.</li> <li data-bbox="488 798 1733 831">4. Installiert und konfiguriert die vordefinierten Services in der Cloud.</li> </ol>	Taxonomie	Lektionen
Kompetenzfeld	Business Engineering		
Objekt	Mögliche Cloud Lösungen im eigenen Arbeitsumfeld.		
Niveau	1		
Voraussetzungen	Keine		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt die Definition und die Vorteile von Cloud Computing (On-Demand Delivery, On-Demand Pricing, Dynamische Skalierung und die Globale Infrastruktur der Hyperscaler).	3	2
1.2	Kennt die Cloud Service- (IaaS, PaaS, SaaS) und Betriebsmodelle (On-Premise, hybrid Cloud und Cloud-Native) und deren Unterschiede.	3	2
1.3	Kennt Computer-Optionen und deren Unterschiede und Anwendungszwecke (virtuelle Maschinen, Container und Serverless).	3	2
1.4	Kennt Cloud Services, um die drei Hauptdatentypen (Strukturierte, teilweise strukturierte und unstrukturierte Daten) zu speichern.	3	2
1.5	Kennt die Verteilung der Verantwortlichkeiten für den sicheren Betrieb von Cloud Services und die grundlegenden Richtlinien und Datenschutzgarantien der grossen Cloud Provider.	3	2
1.6	Kennt das grundlegende Netzwerkkonzept einer Cloud (Virtuelle Private Cloud, Anbindungsmöglichkeiten, Routing, Netzwerksicherheit) und dessen Funktionsweise.	4	2
1.7	Kennt Infrastruktur, Best Practices und weitere Informationsquellen von Hyperscaler (z.B. Referenzarchitekturen und Security Dokumentationen).	4	2
2.1	Kennt Kostenmodelle und das Vorgehen für einfache Kostenanalysen anhand von Beispielen.	4	4
2.2	Kennt die Grundsätze für Cloudmigrationen (z.B. Lift-and-Shift und Lift-and-Reshape).	4	2
3.1	Kennt die Begriffe Verfügbarkeit, Datensicherheit und –resilienz und die Art und Weise, wie deren Anforderungen mit Hochverfügbarkeit, Disaster Recovery und Backup adressiert werden.	3	4
3.2	Kennt die Schritte für die Entwicklung einer einfachen Gesamtarchitektur und definiert die dafür relevanten Cloud Services.	3	6
3.3	Kennt ein Testkonzept zur Überprüfung der Funktionalität, Performance und Sicherheit von Systemen und Diensten.	3	2
4.1	Kennt die Implementationsmöglichkeiten der ausgewählten Dienste in der Cloud (gemäss technischem Konzept).	3	4

### Modul 322 im 3. Semester

Titel		Datenbanksystem in Betrieb nehmen			
Kompetenz	Entwirft und implementiert Benutzerschnittstellen für eine Applikation. Beachtet dabei Standards und ergonomische Anforderungen.				
Handlungsziele				Taxonomie	Lektionen
	1.	Analysiert Benutzereigenschaften und Nutzungsumfeld bezüglich der mit dem System zu lösender Aufgabe und dokumentiert sie		3	4
	2.	Entwirft Varianten einer Benutzerschnittstelle (Maske und Abfolge) anhand vorgegebener Standards und Ergonomieanforderungen.		3	10
	3.	Implementiert eine Benutzerschnittstelle gemäss Entwurf und überprüft problematische Teile auf Machbarkeit.		3	10
	4.	Überprüft eine Benutzerschnittstelle auf Ergonomie.		3	8
	5.	Implementiert eine Benutzerschnittstelle barrierefrei und überprüft sie.		3	8
Kompetenzfeld	Application Engineering				
Objekt	Benutzerschnittstelle zur Interaktion mit einer Applikation, um sein Ziel in gegebenem Umfeld effizient zu erreichen. Zum Beispiel: Interaktive Blogapplikation mit mehreren Masken entwerfen.				
Anzahl Lektionen	40				
Anerkennung	Eidg. Fähigkeitszeugnis				



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt Prozesse (z.B. DesignThinking, UCD) und Methoden (z.B. Beobachtung, Interview, Fokusgruppe, Loganalyse) zur Erhebung und Dokumentation (z.B. Persona, Userstory, Storyboard, Cardsort) von Benutzereigenschaften, Nutzungsumfeld und Aufgabe (Information und Funktion).	3	4
2.1	Kennt Regeln für die Gestaltung einer Benutzerschnittstelle (z.B. Gestalt-Gesetze, Styleguide) sowie Methoden, um diese nach ergonomischen Kriterien (z.B. BestPractices, ISO 9241/16982/25010) aufzubauen.	3	2
2.2	Kennt Möglichkeiten zum effizienten iterativen Entwurf von Benutzerschnittstellen (z.B. Wireframe, PaperPrototype).	3	2
2.3	Kennt Interaktionselemente und entsprechende Konventionen für die Gestaltung von Benutzerschnittstellen (z.B. Menu-Darstellung, Navigation, Interaktion, Orientierung, Ordnung, Benennung).	3	2
2.4	Kennt Regeln zum Aufzeigen der Pflichtfelder und Vorgabe von erwarteten Eingabeformaten.	3	2
2.5	Kennt Möglichkeiten zur gezielten Integration von Hilfefunktionen und Feedbackinformationen in einer Benutzerschnittstelle.	3	2
3.1	Kennt Komponenten (Widgets/Controls) zur Umsetzung der Informations- und Interaktionselemente von Benutzerschnittstellen.	3	5
3.2	Kennt Kriterien, um schwierige Teile der Benutzerschnittstelle bezüglich Benutzerfreundlichkeit und Machbarkeit zu identifizieren und zu analysieren/testen (z.B. clickable Prototype)	3	5
4.1	Kennt Methoden zum effizienten Testen von Benutzerschnittstellen bezüglich Benutzerfreundlichkeit (z.B. Walkthrough, Expertreview, A/B, Usabilitytest).	3	4
4.2	Kennt Metriken zur Beurteilung von Benutzerschnittstellen bezüglich Benutzerfreundlichkeit (z.B. SUS, HEART, Logfile, Umfrage).	3	4
5.1	Kennt die wichtigsten Anforderungen an eine barrierefreie Benutzerschnittstelle (z.B. WCAG, WAI-Aria).	3	4
5.2	Kennt Methoden zum Testen von Benutzerschnittstellen bezüglich Barrierefreiheit (z.B. Screen-Reader, Checklisten)	3	4



## Modul 165 im 4. Semester

Titel		NoSQL-Datenbanken einsetzen		
Kompetenz	Wählt einen geeigneten digitalen Datenspeicher, wenn ein relationales Datenbankmanagementsystem nicht eingesetzt werden soll oder kann. Implementiert eine Datenbank auf einem NoSQL (Not only SQL) Datenbankmanagementsystem, importiert Daten und verwaltet diese.			
Handlungsziele			Taxonomie	Lektionen
	1.	Wählt eine für den Anwendungsfall geeignete NoSQL-Datenbank aus (z.B. Document-Store, Graphen-basiert, Key-Value-Store, Wide-Column-Store, Objekt-orientiert, in-Memory).	4	6
	2.	Implementiert eine NoSQL-Datenbank, befüllt sie mit Daten. Erstelltes Konzept auf Machbarkeit prüfen und gegebenenfalls überarbeiten.	3	6
	3.	Definiert Zugriffsberechtigungen und setzt diese in der NoSQL-Datenbank um.	3	6
	4.	Sichert eine NoSQL-Datenbank in einem Backup und prüft die Wiederherstellung	3	8
	5.	Skaliert eine NoSQL-Datenbank, z.B. durch Replikation.	3	6
	6.	Nutzt die NoSQL-Datenbank lesend und schreibend aus einer Anwendung.	3	8
Kompetenzfeld	Data Management			
Objekt	Mindestens zwei Arten von NoSQL-Datenbanken			
Anzahl Lektionen	40			
Anerkennung	Eidg. Fähigkeitszeugnis			



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt Einsatzzwecke von NoSQL-Datenbanken (z.B. Caching, Datenanalyse, Datawarehouse, Full-Text Search, Netzwerke, Testing).	3	2
1.2	Kennt das CAP-Theorem (Consistency, Availability, Partition tolerance), wonach verteilte Datenbanksysteme charakterisiert werden.	3	2
1.3	Kennt das BASE Konsistenzmodell und die Unterschiede zum ACID-Modell der relationalen Datenbanksysteme.	3	1
1.4	Kennt unterschiedliche Indexing Strukturen, welche in NoSQL-Datenbanken zur Anwendung kommen.	3	1
2.1	Kennt Elemente eines NoSQL-Datenbankmanagementsystems (DBMS) zur Umsetzung eines Datenmodells in ein physisches Datenbankschema.	3	2
2.2	Kennt Befehle einer Datendefinitionssprache (DDL) zur Implementierung des Datenmodells in einem physischen Datenbankschema einer NoSQL-Datenbank.	3	2
2.3	Kennt Befehle einer Datenbearbeitungssprache (DML) zum Hinzufügen einzelner oder mehrerer Datensätze in die NoSQL-Datenbank (über Konsole, aus bestehenden Datensätzen, über DBMS-Frontend-Tools).	3	2
3.1	Kennt Befehle einer Datenkontrollsprache (DCL), um Benutzer und/oder Rollen zu verwalten.	3	3
3.2	Kennt Befehle einer Datenkontrollsprache (DCL), um Zugriffsberechtigungen an Benutzer und/oder Rollen zuzuweisen und zu verwalten.	3	3
4.1	Kennt Befehle eines NoSQL-Datenbankmanagementsystems (DBMS) zur Sicherung einer Datenbank (Daten und Datenbankschema) im ruhenden und laufenden Betrieb.	3	4
4.2	Kennt Befehle eines NoSQL-Datenbankmanagementsystems (DBMS) zur Wiederherstellung einer Datenbank (Daten und Datenbankschema) oder Teile davon.	3	4
5.1	Kennt Unterschiede zwischen verschiedenen Replikationsarten: Multimaster, Master-Slave, Aktiv-Passiv und horizontale Skalierung.	3	3
5.2	Kennt Befehle eines NoSQL-Datenbankmanagementsystems zur Verwaltung der Replikation.	3	3
6.1	Kennt Schnittstellen eines NoSQL-Datenbankmanagementsystems zur Anwendung aus einer Applikation.	3	4

Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
6.2	Kennt den Unterschied zwischen Queries und Projections bei Abfragen von NoSQL-Datenbanken.	3	2
6.3	Kennt den MapReduce Algorithmus zur parallelen Verarbeitung in grossen Datenbeständen einer NoSQL-Datenbank.	3	2

## Modul 347 im 4. Semester

Titel		Dienst mit Container anwenden	
Kompetenz	Erkennt Unterschiede der Containeranwendungsmöglichkeiten im eigenen beruflichen Alltag. Entwickelt mittels einer geeigneten Containerkomposition eine ICT-Lösung für den eigenen Betrieb oder für das betriebliche Umfeld.		
Handlungsziele		Taxonomie	Lektionen
	1. Identifiziert Auswirkungen von Virtualisierung auf den beruflichen Alltag.	3	4
	2. Wählt eine geeignete Containerkomposition (Architektur) situationsbezogen aus.	3	4
	3. Wählt geeignete Containerdienstleister gemäss Anforderungen aus.	3	4
	4. Virtualisiert eine Applikation mit der gewählten Containerkomposition sowohl zu Entwicklungszwecken als auch für Auslieferung/Betrieb.	3	22
	5. Plant Methoden zur Qualitätskontrolle und setzt diese um.	3	6
Kompetenzfeld	Application Engineering		
Objekt	Virtualisierung einer Applikation mit Container.		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		

Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt die wichtigsten Grundlagen, Eigenschaften und Methoden von Container-Technologien (z.B. Verfügbarkeit/Redundanz, Idempotenz, Sicherheit, Ressourcen, Wartbarkeit) sowie deren Auswirkungen auf den beruflichen Alltag.	3	2
1.2	Kennt die Anforderungen des Betriebes auf Bandbreite, Hardware (CPU, RAM, etc.), Instanzen, Skalierbarkeit in Bezug auf Containerisierung.	3	2
2.1	Kennt Architekturformen (z.B. Aufteilung der Dienste auf Container) und deren Aufbau, Konzept und Unterschiede für den Einsatz von Containerlösungen.	3	1
2.2	Kennt die Chancen und Risiken im Umgang mit Containerisierung bezüglich Entwicklung und Betrieb von Diensten und Applikationen	3	1
2.3	Kennt die Anforderungen von Netzwerk, Security etc. in Bezug auf Containerisierung.	3	1
2.4	Kennt die Anwendungsmöglichkeiten von Containern, deren Kompositionen und deren Services	3	1
3.1	Kennt die unterschiedlichen Orchestration Frameworks und Tools (z.B. Docker, Kubernetes) zum Betrieb von Containerplattformen.	3	2
3.2	Kennt globale Anbieter von Containerrepositories und -Plattformen, die Preisunterschiede und die Providerspezifischen Vor- und Nachteile (z.B. Docker-Hub, Azure, AWS etc.).	3	2
4.1	Kennt die Einsatzmöglichkeiten und Vorteile einer containerbasierten Architektur oder Dienstes bezüglich der Bedürfnisse der eigenen Unternehmung.	3	6
4.2	Kennt Konfigurations- und Sicherheitskonzepte einer Container-Komposition.	3	8
4.3	Kennt die Entwicklungsschritte zur Virtualisierung einer Applikation oder Dienstes mit Containern.	3	8
5.1	Kennt Methoden zur Umsetzung, Qualitätssicherung und Qualitätskontrolle von containerisierten Architekturen.	3	2
5.2	Kennt Methoden zur Qualitätskontrolle im Einsatz von mehreren Containern (Kompositionen) innerhalb deren Abgrenzung.	3	2
5.3	Kennt Methoden zur Beweisführung der Testabdeckung, Qualitätssicherheit, Dokumentation und Anwendung.	3	2



## Modul 426 im 4. Semester

Titel	Software mit agilen Methoden entwickeln			
Kompetenz	Agile Methoden zur Softwareentwicklung in Release-Zyklen anwenden.			
			Taxonomie	Lektionen
Handlungsziele	1.	Vorgegebene Funktionalität im Rahmen eines Softwareprojekts mit einer agilen Methode umsetzen.	3	6
	2.	Funktionalitäten schrittweise mit Praktiken der agilen Entwicklung in vorgegebenen Release-Zyklen realisieren, testen und den Software-Release in Kurzform präsentieren.	3	8
	3.	Bestehende Entwurfsmuster und/oder geprüfte Softwarekomponenten zur Lösung von Problemen gezielt einsetzen.	3	6
	4.	Die Ergebnisse und das Arbeitsvorgehen eines Release-Zyklus reflektieren und daraus Erkenntnisse für das weitere Vorgehen ableiten.	3	8
	5.	Die Projektdokumente und den Programmquelltext in einem gemeinsamen Versionsverwaltungssystem bereitstellen.	3	6
	6.	Programmquelltext nach Konventionen intuitiv verständlich formulieren und bei Bedarf überarbeiten.	3	6
Kompetenzfeld	Application Engineering			
Objekt	Funktionsfähiger, getesteter und dokumentierter Software-Release.			
Anzahl Lektionen	40			
Anerkennung	Eidg. Fähigkeitszeugnis			



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt eine agile Methode zur Softwareentwicklung.	3	2
1.2	Kennt Inhalt und Aufbau einer User-Story.	3	1
1.3	Kennt Werte und Prinzipien von agilen Methoden.	3	1
1.4	Kennt die Abgrenzung von agilen und linearen Methoden.	4	2
2.1	Kennt den Ablauf und die Bestandteile eines Releasezyklus.	3	1
2.2	Kennt den Aufbau und Inhalt einer Releasepräsentation.	3	2
2.3	Kennt die Bedeutung von lauffähigen Software Releases.	4	1
2.4	Kennt Praktiken (best practices) der agilen Softwareentwicklung wie z.B. einfaches Design, Metapher, Refactoring, Testdriven Development, Continuous Integration, Pair Programming, Coding Standards.	3	2
2.5	Kennt auf die agile Entwicklung zugeschnittene Tools für Unit-Tests, System-Build und -integration, Akzeptanztest, usw.	4	2
3.1	Kennt die Bedeutung und den Aufbau relevanter Entwurfsmuster.	4	2
3.2	Erkennt den Nutzen der Wiederverwendung bestehender Softwarekomponenten.	3	2
3.3	Kennt ein Vorgehen zur Evaluation von Softwarekomponenten um Problemstellungen zu lösen.	3	2
4.1	Kennt ein Vorgehen um die Ergebnisse eines Releasezyklus zu reflektieren.	3	2
4.2	Kennt Möglichkeiten um das eigene Arbeitsvorgehen im Releasezyklus zu reflektieren.	3	2
4.3	Kennt Möglichkeiten um das weitere Vorgehen aufgrund der Reflektion zu optimieren (SCRUM, ceremony, product backlog issues).	3	4
5.1	Kennt Vorteile und Nutzen eines gemeinsam genutzten Versionsverwaltungssystems.	3	2
5.2	Kennt die Funktionalitäten und die Bedienung eines Versionsverwaltungssystems, sowie die Möglichkeiten unterschiedliche Objekte einzupflegen.	3	2
5.3	Kennt die Unterschiede zwischen Versionsverwaltungssystemen mit exklusivem und parallelem Zugriff.	3	2



<b>Handlungsziele</b>	<b>Handlungsnotwendige Kenntnisse</b>	<b>Taxonomie</b>	<b>Anzahl Lektionen</b>
6.1	Kennt die Konzepte, Regeln und Verfahren von Clean Code.	3	3
6.2	Kennt die Wichtigkeit des Refactoring zur Verbesserung der intuitiven Lesbarkeit von Programm-quelltext.	3	3

## Modul 254 im 4. Semester

Titel		Geschäftsprozesse im eigenen Berufsumfeld beschreiben			
Kompetenz	Dokumentiert Geschäftsprozesse, führt eine Aufgabenanalyse durch und stellt Prozessabläufe grafisch dar.				
				Taxonomie	Lektionen
Handlungsziele	1.	Identifiziert und dokumentiert relevante Prozessinformationen (Bezeichnung, auslösendes Ereignis, Ergebnis, Auslöser, Empfänger) und stützt sich dabei auf die Beschreibung einer zuvor durchgeführten Erhebung ab.		3	14
	2.	Zerlegt Geschäftsprozess in einer Aufgabenanalyse in Prozessschritte und stellt den Prozessablauf unter Berücksichtigung von anerkannten und üblichen Beschreibungsstandards grafisch dar.		3	8
	3.	Präzisiert den Prozessablauf mit Zusatzinformationen wie benötigten Hilfsmitteln, ausführenden Stellen sowie der Beschreibung des zu erzeugenden Outputs (Ergebnis).		3	8
	4.	Teilt Geschäftsprozesse in technische und organisatorische Prozesse auf (Systemdesign).		3	10
Kompetenzfeld	Business Engineering				
Objekt	Bestehender Geschäftsprozess, der alle Grundformen von Folgebeziehungen und 10-20 Prozessschritte umfasst.				
Anzahl Lektionen	40				
Anerkennung	Eidg. Fähigkeitszeugnis				



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt den Unterschied zwischen der Ablauf- und der Aufbauorganisation einer Unternehmung.	3	2
1.2	Kennt die relevanten Elemente einer Unternehmensorganisation, welche die Zuständigkeiten und Verantwortungen in einem Geschäftsprozess beschreiben.	3	3
1.3	Kennt die Grundelemente Objekt und Verrichtung zur Beschreibung eines Geschäftsprozesses.	3	2
1.4	Kennt die wesentlichen Elemente zur Darstellung eines Geschäftsprozesses.	3	3
1.5	Kennt die relevanten Informationen zur Definition eines Geschäftsprozesses und dessen Abgrenzung zu anderen Geschäftsprozessen.	3	2
1.6	Kennt typische Merkmale von auslösenden Ereignissen und Ergebnissen in einem Geschäftsprozess.	4	2
2.1	Kennt eine Möglichkeit, Folgebeziehungen und deren Sachverhalt visuell darzustellen.	4	4
2.2	Kennt eine Darstellungstechnik (z.B. UML Anwendungsfalldiagramm, Aktivitätsdiagramm) zur Modellierung eines Prozessablaufs.	4	4
3.1	Kennt prozessunterstützende Mittel und Ressourcen sowie deren Anwendung (z.B. Checkliste, Workflow-Managementsoftware).	4	4
3.2	Kennt Anforderungen an Informationen und deren Darstellung zur Übergabe an nachfolgende Prozessschritte.	4	4
4.1	Kennt Möglichkeiten, wie Geschäftsprozesse (lösungsneutrale Prozessbeschreibungen) in technische und organisatorische Abläufe aufgeteilt werden können.	3	2
4.2	Kennt Möglichkeiten, um ein Anwendungssystem in organisatorische Systeme und technische Systeme aufzuteilen.	3	4
4.3	Kennt die dadurch neu entstehenden Schnittstellen und deren Zuordnung zu den Teilsystemen.	3	4

## Modul 323 im 5. Semester

Titel	Funktional Programmieren		
Kompetenz	Algorithmen und Teile von Applikationen deklarativ beschreiben und funktional implementieren.		
		Taxonomie	Lektionen
	1. Analysiert und beschreibt Anforderungen, so dass eine Realisierung in funktionaler Programmierung erfolgen kann	3	5
	2. Implementiert Algorithmen und Teilprobleme von Applikationen effizient nach dem funktionalen Programmierparadigma gemäss vorgegebenen Anforderungen.	3	20
	3. Verbessert und optimiert bestehenden imperativ implementierten Code durch Anwendung von funktionaler Programmierung (Refactoring).	3	10
	4. Überprüft Implementierung auf Korrektheit und Qualität	3	5
Kompetenzfeld	Application Engineering		
Objekt	Anwendung mit 3-5 funktionalen Einheiten		
Anzahl Lektionen	40		
Anerkennung	Eidg. Fähigkeitszeugnis		



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt den Unterschied zwischen imperativer und deklarativer/funktionaler Programmierung (deklaratives Programmierparadigma).	3	1
1.2	Kennt Vorgehensweisen zur deklarativen Beschreibung von Problemen und Endzuständen.	3	1
1.3	Kennt Vor- und Nachteile funktionaler Programmierung.	3	1
1.4	Kennt die Begriffe aus dem funktionalen Programmieren (z.B. Pure-Function, Immutable Data, Lambda-Expression, Function, Closure, Callback, Functor) und versteht diese, um sie korrekt anzuwenden.	3	1
1.5	Kennt Möglichkeiten zur Erarbeitung eines für die funktionale Programmierung geeigneten Realisierungskonzeptes.	3	1
2.1	Kann funktionalen Code lesen, verstehen, warten und erweitern.	3	10
2.2	Kennt funktionale Programmierkonzepte (z.B. Filter-Map-Reduce) und passende Entwurfsmuster (z.B. Builder-Pattern)	3	5
2.3	Kennt Wege um funktionalen Code verteilt, parallel und nebenläufig auszuführen.	3	5
3.1	Kennt Mittel um Programmteile daraufhin zu untersuchen, ob diese durch funktionale Implementierung optimiert oder verbessert werden können.	3	5
3.2	Kennt die funktionalen Elemente einer Programmiersprache und kann so Applikationen mit imperativen und deklarativen Programmierparadigma entwickeln.	3	5
4.1	Kennt Mittel, um funktional implementierten Code zu testen (z.B. Unit-Tests).	3	2
4.2	Kennt geeignete Coderichtlinien und hält diese konsequent ein.	3	2
4.3	Kennt Best-Practices für funktionale Programmierung und kann diese anwenden.	3	1



## Modul 450 im 5. Semester

Titel		Applikationen testen		
Kompetenz	Erstellt anhand einer Testbasis ein Testkonzept und leitet daraus Testfälle ab. Implementiert und dokumentiert diese, definiert Korrekturmassnahmen und überprüft Schnittstellen anhand von Sicherheitsvorgaben.			
			Taxonomie	Lektionen
Handlungsziele	1.	Erstellt ein Testkonzept anhand von einem Praxisbeispiel inklusive Anforderungen (Testbasis).	3	6
	2.	Beschreibt ein Testumfeld umfassend.	3	4
	3.	Definiert innerhalb verschiedenen Teststufen/-arten (Unit Tests, Integrationstests, E2E-Tests, Systemtests, Last-/ Performancetests, Security-Tests, Benutzerakzeptanztests, Konformitätstests) Tests und deren Testmittel.	3	5
	4.	Erarbeitet Vorschläge zur Verbesserung von Mängeln innerhalb von Code Reviews.	3	5
	5.	Beschreibt aufgrund von Anforderungen wiederholbare Testfälle inklusive der erwarteten Resultate.	3	5
	6.	Implementiert und führt automatisierte Testfälle aus und dokumentiert dessen Resultate nachvollziehbar.	3	5
	7.	Definiert anhand von beobachteten/gegebenen Abweichungen Korrekturmassnahmen und setzt diese um (z.B. innerhalb Test Driven Development TDD).	3	4
	8.	Testet gemäss Sicherheitskonzept Schnittstellen.	3	6
Kompetenzfeld	Testing			
Objekt	Applikation mit wenigen Features/Userstories für je eine kleine Anzahl Testfälle.			
Anzahl Lektionen	40			
Anerkennung	Eidg. Fähigkeitszeugnis			





Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt die notwendigen Inhalte eines Testkonzepts wie Testziele, Testobjekte, Testarten, Testinfrastruktur, Testorganisation, Testplan.	3	2
1.2	Kennt die Systemgrenzen zur Abgrenzung der im Testkonzept beschriebenen Umgebung.	3	2
1.3	Kennt Strategien zur Klassifizierung von Mängeln.	3	2
2.1	Kennt die wesentlichen Komponenten einer Testumgebung.	3	1
2.2	Kennt den Unterschied und die Gemeinsamkeiten einer Test- und Produktivumgebung.	3	1
2.4	Kennt die wichtigsten Darstellungsarten für die erhobenen Daten und weiss wie diese interpretiert werden.	3	2
3.1	Kennt Beispiele zu den verschiedenen Testarten.	3	1
3.2	Kennt den Einsatz verschiedener Testarten.	3	1
3.3	Kennt das V-Model und die Testpyramide beim Testing.	3	1
3.4	Kennt verschiedene Testmittel.	3	1
3.5	Kennt Möglichkeiten zum Umgang mit Abhängigkeiten (z.B. Interfaces, Injection, Mock, Spy, Stub, Dummy, Fake, Fixtures).	3	1
4.1	Kennt Ansätze zur statischen Überprüfung von Code und Anforderungen.	3	1
4.2	Kennt Clean Code Prinzipien (z.B. DRY, KISS, YAGNI Namensgebung, Modularisierung, SRP) zur Erstellung von testbarem Code und zur Vermeidung von viel Testcode (z.B. auf Userstory/Verhalten testen anstelle auf Implementationsdetails).	3	2
4.3	Kennt Codeverbesserungsmöglichkeiten, die innerhalb eines Code Reviews vorgeschlagen und umgesetzt werden können.	3	1
4.4	Kennt den Unterschied zwischen Testing und Qualitätssicherung.	3	1
5.1	Kennt den Aufbau eines Testfalls (z.B. Vorbedingung, Eingabewerte, Resultat, Nachbedingung, positiv- und negativ/Fehler-Test).	3	2
5.2	Kennt Kriterien, um einen Test wiederholbar zu machen (Retesting, Regression, Testdaten, Testbenutzer/Berechtigungen).	3	2
5.3	Kennt den Unterschied zwischen manuellen und automatisierten Tests.	3	1



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
6.1	Kennt Testframeworks, um automatisierte Tests zu definieren (Schritte Given/Arrange, When/Act, Then/Assert) und durchzuführen.	3	2
6.2	Kennt Methoden, um Testresultate nachvollziehbar zu dokumentieren (z.B. Traceability Tools).	3	2
6.3	Kennt die wesentlichen Inhalte eines Testprotokolls: Tester, Testobjekt, Testzeitpunkt, Testumgebung (z.B. Browser), Testresultat.	3	1
7.1	Kennt Debugging und dessen Anwendung.	3	2
7.2	Kennt den TDD Ansatz ausreichend, um ihn an einem einfachen Beispiel anzuwenden.	3	2
8.1	Kennt einfache Methoden zum Testen von Schnittstellen.	3	6

## Modul 183 im 6. Semester

Titel		Applikationssicherheit implementieren			
Kompetenz		Applikationen sicher planen, entwickeln und in Betrieb nehmen.			
				Taxonomie	Lektionen
Handlungsziele	1.	Aktuelle Bedrohungen erkennen und erläutern können. Aktuelle Informationen zum Thema (Erkennung und Gegenmassnahmen) beschaffen und mögliche Auswirkungen aufzeigen und erklären können.		3	8
	2.	Sicherheitslücken und ihre Ursachen in einer Applikation erkennen können. Gegenmassnahmen vorschlagen und implementieren können.		3	8
	3.	Mechanismen für die Authentifizierung und Autorisierung umsetzen können.		3	8
	4.	Sicherheitsrelevante Aspekte bei Entwurf, Implementierung und Inbetriebnahme berücksichtigen.		3	8
	5.	Informationen für Auditing und Logging generieren. Auswertungen und Alarme definieren und implementieren.		3	8
Kompetenzfeld	Application Engineering				
Objekt	Web-Applikation mit Datenanbindung, Webservice.				
Anzahl Lektionen	40				
Anerkennung	Eidg. Fähigkeitszeugnis				



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt Informationsquellen zu aktuellen Bedrohungen.	3	4
1.2	Kennt Sicherheitslücken und mögliche Folgen von Angriffen.	3	4
2.1	Kennt mögliche Ursachen von Sicherheitslücken in Applikationen.	3	2
2.2	Kennt ein Vorgehen zur Identifikation von Sicherheitslücken in Applikationen.	3	2
2.3	Kennt geeignete Gegenmassnahmen zu den verschiedenen Kategorien von Bedrohungen und wie diese implementiert werden	3	4
3.1	Kennt Authentifizierungsmechanismen und deren Funktionsweise.	3	2
3.2	Kennt Verschlüsselungsmechanismen und deren Einbindung in der Applikation.	3	4
3.3	Kennt verschiedene Verfahren zur Zugriffssteuerung.	3	2
4.1	Kennt sicherheitsrelevante Aspekte beim Entwurf von Applikationen.	3	2
4.2	Kennt Techniken bei der Realisierung zur Vermeidung von Sicherheitslücken (z.B. Input/ Output Validierung, Defensives Programmieren, Session-Management, Error Handling).	3	4
4.3	Kennt ein Vorgehen für das sichere Deployment von Applikationen.	3	2
5.1	Kennt Sinn, Aufbau und Inhalt eines Logs.	3	2
5.2	Kennt Sinn, Aufbau und Inhalt eines Audit-Trails.	3	2
5.3	Kennt mögliche Formen der Alarmierung und Regeln für eine Alarmauslösung.	3	4

## Modul 306 im 6. Semester

Titel		Kleinprojekte im eigenen Berufsumfeld abwickeln		
Kompetenz	Wickelt Kleinprojekte im Team mit klar definierter Zielsetzung, Anforderungen, vorgegebenen Ressourcen und Terminen mit geeigneten Softwaretools ab.			
Handlungsziele		Taxonomie	Lektionen	
	1.	Prüft eine Zielsetzung unter Berücksichtigung der vorgegebenen Ressourcen, Anforderungen und Termine auf Machbarkeit, hält die Erkenntnisse fest und bespricht diese mit dem Auftraggeber.	3	6
	2.	Identifiziert ein Projekt und bestimmt Massnahmen zur Bewältigung anhand eines Business Cases (Stakeholder, Risikoanalyse, Machbarkeits- und Wirtschaftlichkeitsanalyse, Zeitmanagement, Zielentwicklung).	3	4
	3.	Erstellt einen Projektplan zur systematischen Abwicklung eines Auftrags unter Berücksichtigung der Ressourcen, Termine, Problemstellungen und Arbeitsteilung (Wasserfallmethode/agile Methode unabhängig von Methode im Lehrbetrieb).	3	8
	4.	Erteilt Arbeitsaufträge (Arbeitspaket/Stories) und koordiniert und überwacht deren Ausführung (Qualität, Termine, Kosten).	3	8
	5.	Erstellt einen Arbeitsfortschrittsbericht (Projektstatusbericht, Projektjournal/Board) und dokumentiert darin den Stand des Projekts (Ressourcen, Termine, geleistete Arbeiten) für den Auftraggeber.	3	6
	6.	Dokumentiert und erläutert das Projektergebnis (Lösung) für den Auftraggeber.	3	4
	7.	Reflektiert den Projektverlauf mit den Projektbeteiligten im Team und leitet Erkenntnisse ab, wie effiziente Projektarbeit gestaltet werden kann.	3	4
Kompetenzfeld	Project Management			
Objekt	Eigenes Projekt mit definierten Zielen, Anforderungen und Ergebnissen sowie vorgegebenen Ressourcen und Terminen in einem Team von 3-5 Personen.			
Anzahl Lektionen	40			
Anerkennung	Eidg. Fähigkeitszeugnis			



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt Merkmale eines Vorhabens, die dafürsprechen, dieses sinnvoll in Form eines Projektes abzuwickeln (Projektdefinition, Projektarten).	3	2
1.2	Kennt die wichtigsten Rahmenbedingungen und deren Einfluss, die hinsichtlich der zeitlichen Machbarkeit eines Auftrages zu berücksichtigen sind (z.B. Anzahl Mitarbeitende, Verfügbarkeit der Mitarbeitenden, Anzahl Arbeitstage bis zum Endtermin, Verfügbarkeit der notwendigen Sachmittel).	3	2
1.3	Kennt die wichtigsten Rahmenbedingungen und deren Einfluss, die hinsichtlich der inhaltlichen Machbarkeit eines Auftrages zu berücksichtigen sind (z.B. Kompetenzen der Mitarbeitenden, technologische Voraussetzungen).	3	2
2.1	Kennt die Analysemöglichkeiten anhand des Business Cases und Anwendungsbeispiele.	3	4
3.1	Kennt ein Vorgehensmodell zur Aufteilung eines Projektes in Phasen und deren systematische Abwicklung.	3	2
3.2	Kennt die Kriterien zur Wahl des effizientesten Vorgehensmodells (klassisch/agil).	3	2
3.3	Kennt Kriterien zur Bildung von Arbeitsaufträgen/Arbeitspaketen, die unter Berücksichtigung der Arbeitsteilung und Abwicklung in einem Projektteam parallel und sequentiell zu bearbeiten sind.	3	4
4.1	Kennt die Elemente und deren Bedeutung für eine vollständige Abwicklung eines Arbeitsauftrags (z.B. Zielsetzungen, Rahmenbedingungen, Termine, Ressourcen, Skills).	3	4
4.2	Kennt Methoden, um den Projektfortschritt zu überwachen.	3	4
5.1	Kennt die Bedeutung einer regelmässigen Berichterstattung an den Auftraggeber.	3	3
5.2	Kennt Aufbau, Inhalt und Darstellung eines Arbeitsfortschrittsberichts (z.B. Termine, Kosten, Qualität, Nachvollziehbarkeit).	3	3
6.1	Kennt Aufbau, Inhalt und formale Regeln, die bei der Dokumentation der Arbeitsergebnisse zu berücksichtigen sind.	3	4
7.1	Kennt Methoden zur Steuerung und Förderung der Zusammenarbeit (z.B. Abgrenzung der Aufträge, regelmässige Koordination, Feedback einholen/geben).	3	4



## Modul 321 im 7. Semester

Titel		Verteilte Systeme programmieren			
Kompetenz	Verteilte Systeme analysieren, verstehen, planen, erweitern und anwenden, sowie bestehende Applikationen in ein verteiltes System überführen.				
				Taxonomie	Lektionen
Handlungsziele	1.	Analysiert unterschiedlich strukturierte Softwaresysteme und überführt diese in verteilte Systeme.		3	8
	2.	Verwendet Systemkomponenten in verteilten Systemen.		3	10
	3.	Verbindet Systemteile über wohl definierte Schnittstellen.		3	10
	4.	Implementiert Systemkomponenten in einem verteilten System und überprüft deren Funktion.		3	12
Kompetenzfeld	Application Engineering				
Objekt	Entwickeln von Systemkomponenten in verteilten Systemen, wie eine Zeiterfassung, mit der Anbindung von fertigen Systemkomponenten zur Authentifizierung und Autorisierung sowie einem Dienst zur persistenten Datenhaltung.				
Anzahl Lektionen	40				
Anerkennung	Eidg. Fähigkeitszeugnis				



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt Begriffe, die im Bezug mit verteilten Systemen verwendet werden (z.B. Cloud-Native, Cloud-Functions, XaaS).	3	2
1.2	Kennt die Eigenschaften (z.B. Idempotenz/Retry, Resilience/design for failure, Transaktion/CQRS/EventSourcing) von verteilten Systemen (z.B. Rechnerverbund).	3	2
1.3	Kennt Systemarchitekturen zur Softwareentwicklung (z.B. Monolith, Microservices, Client-Server, Peer to Peer).	3	2
1.4	Kennt Vorgehensweisen, um Softwaresysteme zu migrieren (z.B. Strangler-Pattern, Big Bang).	3	2
2.1	Kennt Kategorien von Systemkomponenten (z.B. Identitätsverwaltung, Ereignisverwaltung, Monitoring, Datenhaltung, Lastenverteilung).	3	2
2.2	Kennt aktuelle Technologien für die jeweiligen Kategorien von Systemkomponenten (z.B. KeyCloak, RabbitMQ, LogStash, Prometheus, Redis, Minio, Traefik, Caddy).	3	4
2.3	Kennt Möglichkeiten, um verteilte Systeme lokal zu entwickeln und bestehende Systemkomponenten anzubinden.	3	4
3.1	Kennt die Anwendung des Pub-Sub-Patterns (z.B. Message-Broker wie RabbitMQ, Kafka und Protokolle wie AMQP, MQTT).	3	2
3.2	Kennt Hilfsmittel zum Serialisieren und Deserialisieren von Daten (z.B. JSON, XML, protobuf).	3	2
3.3	Kennt Protokolle (z.B. HTTP, TCP) zum Datenaustausch über wohl definierte Schnittstellen (z.B. REST, GraphQL, gRPC, WS, Gateway, Proxy).	3	4
3.4	Kennt Mechanismen für Authentifizierung und Autorisierung (z.B. OpenID, JWT, OAuth, ABAC/RBAC).	3	2
4.1	Kennt Werkzeuge, um Softwareanforderungen in einem verteilten System zu implementieren (z.B. Container und Virtualisierung, Clouddienste wie CloudFoundry, AWS).	3	2
4.2	Kennt Methoden zur horizontalen und vertikalen Skalierung von verteilten Systemkomponenten.	3	2
4.3	Kennt Mittel um einzelne Systemteile und verteilte Systeme, sowie deren Schnittstellen zu dokumentieren (z.B. OpenAPI, Pact, Pre-/Postcondition & Invariante).	3	4





Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
4.4	Kennt Vorgehensweisen, um die Funktionstüchtigkeit von verteilten Systemen zu überprüfen (z.B. Health-Check, Situationsanalyse, automatisierte und manuelle Tests) und Fehler einzugrenzen.	3	4



**Modul 324 im 7. Semester**

Titel		DevOps-Prozesse mit Tools unterstützen		
Kompetenz	Setzt automatisierende Tools für die verschiedenen Prozessschritte der Applikationsentwicklung ein ("Continuous Intergation/Continuous Deployment").			
		Taxonomie	Lektionen	
Handlungsziele	1.	Erfasst und verwaltet die Anforderungen und Umsetzungsschritte nachvollziehbar für die Entwicklung im Team	3	4
	2.	Konfiguriert und nutzt die lokale Entwicklungsumgebung integriert mit den automatisierenden Tools.	3	8
	3.	Wendet die Sourcecode- und Artefaktverwaltung für eine anforderungsorientierte Entwicklung im Team an	3	4
	4.	Nutzt einen automatisierten Integrationsprozess (Build, Qualitätssicherung/Test der Artefakte)	3	12
	5.	Nutzt einen automatisierten Deploymentprozess (Installation und Konfiguration der Laufzeitumgebung und Applikation)	3	12
Kompetenzfeld	Application Engineering			
Objekt	Einfache framework/library-basierte Webapplikation. Zum Beispiel: ToDo-Liste-Applikation automatisiert von Spezifikation und Entwicklung über Testing auf Abnahmeumgebung bringen.			
Anzahl Lektionen	40			
Anerkennung	Eidg. Fähigkeitszeugnis			



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt den Nutzen bezüglich kontinuierlicher toolunterstützter Entwicklung und Wartung (z.B. MVP, Kundenfeedback, Kosten/Nutzen, Qualität, Risikoreduktion).	4	2
1.2	Kennt Vorgehensweisen zur Verwaltung von Anforderungen (z.B. Stories, Issues, Akzeptanzkriterien etc.).	4	1
1.3	Kennt Vorgehensweisen zur nachvollziehbaren Entwicklung im Team (z.B. Verknüpfung Commit mit Story, Pullrequest/PeerReview).	4	1
2.1	Kennt Möglichkeiten, um automatisierende Tools in der Entwicklungsumgebung zu nutzen (z.B. Linting, Build, Ausführen, Testen, Versionierung, Abhängigkeiten/Pakete).	5	8
3.1	Kennt Praktiken, um Sourcecode featurebasiert zu verwalten (z.B. GIT-Workflow, semantic Versioning etc.).	5	2
3.2	Kennt Möglichkeiten, um benötigte und erstellte Artefakte/Binaries zu verwalten (z.B. Artefakt-Repository/Registry, Dependency-Management wie npm/maven etc.).	3	2
4.1	Kennt Möglichkeiten, um die Applikation und Komponenten automatisiert zu builden, auszuführen und zu testen (z.B. Build-Management wie gradle/npm, Container, Pipelines etc.).	3	12
5.1	Kennt Möglichkeiten, um die Applikation automatisiert zu deployen und konfigurieren (z.B. Infrastructure as Code, Scripting, Virtualisierung, Konfiguration mit YAML, Vaults, Profile, SSH-Key etc.).	3	12



## Modul 241 im 8. Semester

Titel		Innovative ICT-Lösungen initialisieren		
Kompetenz	Erarbeitet mögliche innovative Lösungsansätze für eine bestimmte Problemstellung einer Zielgruppe. Präsentiert diese anderen Fachpersonen in Form eines kurzen Pitch.			
			Taxonomie	Lektionen
Handlungsziele	1.	Erklärt einem Arbeitskollegen, weshalb Innovation für das Erstellen langfristiger Marktleistungen bedeutend ist.	3	8
	2.	Identifiziert ein Problem für eine definierte Zielgruppe.	3	10
	3.	Validiert mögliche Lösungsansätze mit Hilfe primärer und sekundärer Marktforschung.	3	6
	4.	Bewertet Lösungsvarianten unter Berücksichtigung technischer und betriebswirtschaftlicher Potentiale und Risiken mit Einbezug der definierten Zielgruppe.	3	10
	5.	Präsentiert die erarbeiteten Lösungsvarianten anderen Fachpersonen in Form eines kurzen Pitch.	3	6
Kompetenzfeld	System Management			
Objekt	Aktuelle Problemstellung aus einem beliebigen Bereich der ICT innovativ lösen, beispielsweise anhand der Methode von <a href="http://www.getkickbox.com">www.getkickbox.com</a> .			
Anzahl Lektionen	40			
Anerkennung	Eidg. Fähigkeitszeugnis			



Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt die Zusammenhänge zwischen Innovation und Wettbewerbsfähigkeit eines Unternehmens.	3	2
1.2	Kennt Umstände, welche Innovation in betrieblichen Strukturen ermöglichen oder behindern.	3	2
1.3	Kennt Beispiele wie auch Gründe von erfolgreichen und gescheiterten Innovationen des aktuellen Marktes.	3	2
1.4	Kennt zielführende Modelle, um innovative Ideen zu steuern.	3	2
2.1	Kennt Möglichkeiten, eine Problemstellung kurz und verständlich zu definieren.	3	4
2.2	Kennt Methoden, passend zur Problemstellung, um Personas und Zielgruppen zu definieren.	3	2
2.3	Kennt Erhebungsmethoden, um relevante Informationen einer Zielgruppe zu erhalten.	3	4
3.1	Kennt den Nutzen von Marktforschung.	3	2
3.2	Kennt den Unterschied primärer und sekundärer Marktforschung.	3	2
3.3	Kennt Methoden, um Informationen aus der Marktforschung auszuwerten und aussagekräftig darzustellen.	3	2
4.1	Kennt Methoden, um Lösungsvarianten zu bewerten.	3	4
4.2	Kennt Methoden, um das Risiko und das Potenzial abzuschätzen.	3	2
4.3	Kennt technische und betriebswirtschaftliche Faktoren, welche für die Wahl der Lösungsvariante relevant sind.	3	4
5.1	Kennt die essenziellen Inhalte eines Pitch und den Unterschied zu herkömmlichen Präsentationstechniken.	3	2
5.2	Kennt Hilfsmittel, um die Lösungsvariante mittels Pitch zu präsentieren.	3	2
5.3	Kennt die Bedeutung einer kurzen und prägnanten Präsentation.	3	2

## Modul 245 im 8. Semester

Titel		Innovative ICT-Lösungen umsetzen			
Kompetenz	Realisiert auf der Basis eines erarbeiteten innovativen Lösungsansatzes einen Proof of Concept. Präsentiert diesen anderen Fachpersonen in Form eines kurzen Pitch.				
				Taxonomie	Lektionen
Handlungsziele	1.	Eruiert die Risiken zur technischen Machbarkeit, sowie geeignete Massnahmen zu deren Minimierung.		3	8
	2.	Erarbeitet einen technischen Prototyp, um einen Lösungsansatz zu validieren.		3	10
	3.	Trägt alle Erkenntnisse zum Lösungsansatz in einem Proof of Concept zusammen.		3	10
	4.	Präsentiert anderen Fachpersonen den Proof of Concept in einem kurzen Pitch.		3	12
Kompetenzfeld	Business Engineering				
Objekt	Aktuelle Problemstellung aus einem beliebigen Bereich der ICT innovativ lösen, beispielsweise anhand der Methode von <a href="http://www.getkickbox.com">www.getkickbox.com</a> .				
Anzahl Lektionen	40				
Anerkennung	Eidg. Fähigkeitszeugnis				

Handlungsziele	Handlungsnotwendige Kenntnisse	Taxonomie	Anzahl Lektionen
1.1	Kennt Methoden zur Risikoidentifikation.	3	4
1.2	Kennt mindestens eine Methode, um Risiken zur technischen Machbarkeit zu beurteilen.	3	2
1.3	Kennt mindestens eine Methode, um Risiken zur technischen Machbarkeit zu minimieren.	3	2
2.1	Kennt die Unterschiede und den Einsatzzweck von Prototyp, Minimum Viable Product (MVP) und Proof of Concept (PoC) und kennt die Vor- und Nachteile.	3	4
2.2	Kennt Varianten, um einen Prototyp mit einfachen Mitteln zu erstellen.	3	4
2.3	Kennt die Bedeutung, die funktionalen Anforderungen zu prüfen.	3	2
3.1	Kennt die Bedeutung eines PoC.	3	4
3.2	Kennt die wesentlichen Inhalte eines PoC.	3	2
3.3	Kennt Vorgehensweisen, um eine PoC zu erstellen.	3	2
3.4	Kennt Vorgehensvarianten, welche auf einen PoC folgen können.	3	2
4.1	Kennt die Bedeutung der Resultate des PoC. Zum Beispiel betriebswirtschaftlicher Nutzen, Potential, gesellschaftliche Bedeutung, etc. und kann dies in einem Pitch einem Fachpublikum erklären.	3	4
4.2	Kennt die essenziellen Inhalte eines Pitch und den Unterschied zu herkömmlichen Präsentationstechniken.	3	2
4.3	Kennt Hilfsmittel, um den PoC mittels Pitch zu präsentieren.	3	4
4.4	Kennt die Bedeutung einer kurzen und prägnanten Präsentation.	3	2

## Mathematik 1. Semester

### Lektionenverteilung über alle Semester:

1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.	7. Sem.	8. Sem.	Total Lektionen
40	40	20	20	0	0	0	0	<b>120 Lektionen</b>

Woche	Lerninhalte	Taxonomie	Methodische Hinweise
1	Zahlen, Runden, Prozente, Zehnerpotenzen	2	
2	Zehnerpotenzen, Einheiten, Multiplikation und Division von Zehnerpotenzen	2	
3	Rechnen mit Zehnerpotenzen gemischte Aufgaben, Addition von Termen	2	
4	Addieren von Termen mit Variablen, Subtrahieren rationaler Zahlen	2	
5	Addieren und Subtrahieren von Termen mit unterschiedlichen Vorzeichen	2	
6	Vermischte Aufgaben zum Addieren und Subtrahieren, Addieren und Subtrahieren mit Klammerausdrücken	2	
7	Gemischte Aufgaben zum Thema Klammerausdrücke, Multiplikation, Multiplikation mit Vorzeichen	2	
8	Multiplizieren mit Summentermen, Zerlegen in Faktoren	2	
9	Vermischte Aufgaben zur Multiplikation, Division von Termen	2	
10	Kürzen von Bruchtermen, Erweitern von Bruchtermen	2	
11	Addieren und Subtrahieren gleichnamiger und ungleichnamiger Bruchterme	2	
12	Dividieren von Bruchtermen, Dividieren von Summentermen	2	
13	Dividieren von Summentermen durch Summenterme, Einführung zum Potenzieren	2	
14	Addition und Subtraktion von Potenzen, Dividieren von Potenztermen	2	
15	Potenzieren von Potenztermen, Potenzieren von Summen und Summentermen	2	
16	Binomische Formeln, Wurzelziehen	2	
17	Addieren und Subtrahieren von Wurzeltermen	2	
18	Multiplizieren von Wurzeltermen, Dividieren von Wurzeltermen	2	
19	Potenzieren von Wurzeltermen	2	
20	Gemischte Aufgaben Wurzeln und Potenzen	2	



## Mathematik 2. Semester

Woche	Lerninhalte	Taxonomie	Methodische Hinweise
1	Gleichungen	2	
2	Gleichungen	2	
3	Gleichungen mit Klammern	2	
4	Gleichungen mit Brüchen	2	
5	Gleichungen mit Brüchen	2	
6	Textgleichungen	2	
7	Textgleichungen	2	
8	Gleichungssysteme	2	
9	Gleichungssysteme	2	
10	Funktionen	2	
11	Funktionen	2	
12	Funktionen am PC – Grafische Darstellung	2	
13	Funktionen am PC – Grafische Darstellung	2	
14	Quadratische Gleichungen	2	
15	Quadratische Gleichungen	2	
16	Quadratische Gleichungen am PC – Grafische Darstellung	2	
17	Quadratische Gleichungen mit quadratischer Ergänzung , p/q- Formel und abc Formel	2	
18	Satz von Vieta	2	
19	Repetition	2	

### Mathematik 3. Semester

Woche	Lerninhalte	Taxonomie	Methodische Hinweise
1	Wurzeln	2	
2	Wurzeln	2	
3	Wurzeln	2	
4	Wurzeln	2	
5	Wurzeln	2	
6	Logarithmen	2	
7	Logarithmen	2	
8	Logarithmen	2	
9	Logarithmen	2	
10	Logarithmen	2	
11	Logarithmen	2	
12	Logarithmen	2	
13	Funktionen - Anwendungsbeispiele	2	
14	Funktionen - Anwendungsbeispiele	2	
15	Funktionen - Anwendungsbeispiele	2	
16	Funktionen - Anwendungsbeispiele	2	
17	Funktionen - Anwendungsbeispiele	2	
18	Flächenberechnungen	2	
19	Flächenberechnungen	2	
20	Flächenberechnungen	2	



## Mathematik 4. Semester

Woche	Lerninhalte	Taxonomie	Methodische Hinweise
1	Flächenberechnungen	2	
2	Volumenberechnungen	2	
3	Volumenberechnungen	2	
4	Volumenberechnungen	2	
5	Volumenberechnungen	2	
6	Volumenberechnungen	2	
7	Strahlensatz	2	
8	Strahlensatz	2	
9	Pythagoras	2	
10	Pythagoras	2	
11	Pythagoras	2	
12	Trigonometrie	2	
13	Trigonometrie	2	
14	Trigonometrie	2	
15	Trigonometrie	2	
16	Trigonometrie	2	
17	Statistik	2	
18	Statistik	2	
19	Statistik	2	

## Englisch 1. bis 8. Semester

### Lektionenverteilung über alle Semester:

1. Sem.	2. Sem.	3. Sem.	4. Sem.	5. Sem.	6. Sem.	7. Sem.	8. Sem.	Total Lektionen
20	20	40	40	20	20	20	20	160 Lektionen

**Lehrmittel:** **Headway (Oxford)** Intermediate Course Book / Workbook Semester 1 - 4  
**INFOTECH for Computer Users** (Cambridge University Press mit Online Übungen) und Fachtexte aus Internet / verschiedenen Fachbüchern zu IT Semester 5 – 8

Pro Unit Headway File werden ca. 6 - 8 Lektionen benötigt (inklusive Lernkontrollen). Hinzu kommen ca. 1 Lektion Besprechen der Lernkontrollen und schriftlichen Arbeiten pro Unit. Es wird vermehrt Wert auf Kommunikation (Sprechen) und Wortschatz gelegt. Die Grammatik ist teilweise passive zu verstehen  
 Ab dem 5. Semester Technical English basierend INFOTECH (Semester 5+6) und auf Fachtexten aus dem IT-Bereich (Semester 7+8). Vorschlag: Fachtexte aus Internet und Fachbüchern, Kurzreferate und Dokumentationen oder Reports verfassen. Pro Unit INFOTECH rechnet man mit ca. 2-3 Lektionen ohne Lernkontrollen.

Semester	Lektionen	Units	Inhalte	Themen
<b>Lehrmittel Headway (Intermediate)</b>				
1	20	1 – 3	<b>Grammar</b> Question words Tenses and auxiliary verbs Passive v. Active  <b>Vocabulary</b> Adjectives Word formation Parts of speech Leisure activities  <b>Listening</b> Family Nationalities Work-life balance  <b>Writing</b> Describing someone Formal and informal Email	<ul style="list-style-type: none"> <li>- Family</li> <li>- Friends</li> <li>- Education</li> <li>- Free-time</li> <li>- Languages</li> <li>- Work life</li> </ul>

Semester	Lektionen	Units	Inhalte	Themen
<b>Lehrmittel Headway (Intermediate)</b>				
			Formal letter	
2	20	4 – 6 + Review	<p><b>Grammar</b> Narrative tenses Modal and related verbs Present perfect</p> <p><b>Vocabulary</b> Homophones Phrasal verbs Verbs and adverbs</p> <p><b>Listening</b> Famous stories Rules and philosophy for life</p> <p><b>Writing</b> Story writing (1) Writing for talking (1) Adverbs in descriptive writing</p>	<ul style="list-style-type: none"> <li>- Famous story-tellers (Shakespeare, Oscar Wilde)</li> <li>- Story-telling (speaking and writing)</li> <li>- Rules and customs</li> <li>- Polite requests and offers</li> </ul>
3	40	7 – 10	<p><b>Grammar</b> Verb patterns Adjectives and prepositions Future forms Second and third conditionals Modals of probability</p> <p><b>Vocabulary</b> Parts of the body Word building Words with similar meanings Phrasal verbs</p> <p><b>Listening</b> Friends The future of energy</p>	<ul style="list-style-type: none"> <li>- Friends</li> <li>- Boarding school</li> <li>- Body language</li> <li>- Artificial intelligence</li> <li>- Energy</li> <li>- Economy / money</li> <li>- Fact or fiction</li> <li>- Wedding</li> </ul>

Semester	Lektionen	Units	Inhalte	Themen
<b>Lehrmittel Headway (Intermediate)</b>				
			<p>The kindness of strangers The man who posted himself to Australia</p> <p><b>Writing</b> Arguing for and against Writing for talking (2) Describing places Story writing (2)</p>	
4	40	<p>11 – 12 + Review</p> <p>+ start with new book</p>	<p><b>Grammar</b> Noun phrases Articles Possessives Reported speech Indirect questions</p> <p><b>Vocabulary</b> Compound nouns Verbs + prepositions</p> <p><b>Listening</b> News Having the time of his life</p> <p><b>Writing</b> Informal language A biography</p>	<ul style="list-style-type: none"> <li>- Digital v. physical</li> <li>- Digital animation</li> <li>- Social media</li> <li>- Job interviews</li> <li>- Clichés</li> </ul>

Semester	Lektionen	Units	Inhalte	Themen
Neues Lehrmittel (INFOTECH for Computer Users)				
5	20	ca. 1 – 7	Vocabulary Word families Writing Emails Revision of grammar	
6	20	ca. 8 – 14	Vocabulary Connectors, prefixes and suffixes Presentations IT related Writing Short texts Revision of grammar	
7	20	ca. 15 – 22	Vocabulary Affixation, conversion and Compounding Writing Short texts Revision of grammar	
8	20	ca. 23 – 30	Vocabulary Collocations, passive constructions Presentations My company, job-related topics Writing Applying for a job, CV writing Revision of grammar	



## Beschreibung der Taxonomiestufen

Kompetenzstufe	Denk- und Arbeitsprozess	Bedeutung
<b>K1: Wissen</b> Informationen wiedergeben und in gleichartigen Situationen abrufen	nennen, aufzählen	Punkte, Gedanken, Argumente, Fakten auflisten
	benennen	Vorgegebenen Elementen den Namen geben.
<b>K2: Verstehen</b> Informationen nicht nur wiedergeben, sondern auch verstehen	bestimmen, definieren	Den Inhalt eines Begriffs auseinanderlegen; feststellen; etwas herauslesen, etwas veranschaulichen.
	Das Grundprinzip von etwas erklären	Die Idee erklären, die einer Sache zugrunde liegt, nach der etwas wirkt; schematisch erklären, wie etwas aufgebaut ist (keine Einzelheiten des inneren Aufbaus, der inneren Abläufe).
	zuordnen	Elemente miteinander in Verbindung bringen, gruppieren
	unterscheiden, vergleichen	Die Unterschiede zwischen Dingen anhand bestimmter Merkmale/Kriterien herausheben.
	beschreiben, erläutern, erklären	Etwas mit eigenen Worten deutlich machen, darstellen, kennzeichnen, treffend schildern (z.B. indem „W-Fragen“ beantwortet werden).
<b>K3: Anwenden</b> Informationen über Sachverhalte in verschiedenen Situationen anwenden	anwenden	Bei einer Arbeit ein bestimmtes Verfahren, eine bestimmte Technik zu einem bestimmten Zweck verwenden. Wissen, Begriffe, Konzepte, Modelle umsetzen um gewohnte, bekannte Anforderungen zu bewältigen.
	ausführen, durchführen	Ein Vorhaben in allen Einzelheiten verwirklichen, eine bestimmte Arbeit erledigen, fachgerecht in die Praxis umsetzen.
	lokalisieren	Örtlich auffinden; den Ort, die Lage von etwas bestimmen.
	instand halten, warten	In brauchbarem Zustand halten. Arbeiten ausführen, die für die Funktionsfähigkeit periodisch nötig sind. Bauteile oder Systeme austauschen.
	Instand setzen, reparieren	Bauteile oder Systeme reparieren.
	berechnen	Mit Hilfe üblicher Angaben, dem Formelbuch und Taschenrechner praxismgerechte Antworten auf branchenspezifische Fragestellungen geben. Nur Formeln anwenden, keine Formeln umstellen oder entwickeln.
	befolgen	Sich nach etwas richten (z. B. nach einer Vorschrift handeln). Informationen über Sachverhalte in verschiedenen Situationen anwenden.
<b>K4: Analyse</b> Sachverhalte in Einzelelemente gliedern, die Beziehungen zwischen Elementen aufdecken und Zusammenhänge erkennen	kommentieren	Einen Befund abgeben zu Theorien, Anforderungen, Situationen, zur Beschaffenheit eines Gegenstandes. Dies erfolgt durch Erläuterung, Auslegung, kritische Stellungnahmen.
	beraten	Bei einem komplexen, theoretischen Phänomen oder einer praktischen Problemstellung, mit Rat beistehen bzw. Ratschläge geben.
	begründen	Etwas breit und tief und von verschiedenen Standpunkten aus prüfen, auslegen, nachweisen, deutlich machen; dazu Gründe und Argumente hervorheben.
<b>K5: Synthese</b> Elemente eines Sachverhalts kombinieren u. zu einem Ganzen zusammenfügen od. eine Lösung für Probleme entwerfen.	situationsgerecht umgehen, optimieren, geeignete Massnahmen ableiten	Einzelne Elemente eines Sachverhalts, einer Situation, zu einer neuen Lösung zusammenfügen. Die bestmögliche Lösung eines neuen Problems finden und in die Praxis umsetzen.
	zeichnen, aufzeichnen	Etwas (Ganzes und Teile) bildhaft darstellen. Die Wirklichkeit mit Hilfe von Normen abbilden. Ein Gegenstand als Handskizze darstellen.
<b>K6: Bewerten</b> Bestimmte Gegenstände, Informationen und Sachverhalte nach Kriterien beurteilen	prüfen	Der Zustand und die Funktion gewisser Elemente anhand von Kriterien untersuchen. Daraus ein Urteil ableiten.
	beurteilen, diagnostizieren, ableiten	Gegenstände, Sachverhalte, Phänomene, Lösungen anhand von Kriterien beurteilen (Kriterien können sein: Zustand, Aussehen, einwandfreies Funktionieren, ...). Aus dem Urteil eine Lösung, Empfehlung oder Entscheidung ableiten.
	interpretieren	Die Bedeutung von etwas erklären, die Kernaussagen herauschälen, mit einer persönlichen Beurteilung verknüpfen.