Mirocredentials in ESCO und Europass

Prof. Kay Berkling, PhD
Duale Hochschule Baden-Württemberg (DHBW)
DAAD Frankfurt
4.12.2025

Definition Microcredentials (MC)

Micro-Credentials sind kleine, klar abgegrenzte Lerneinheiten, die spezifische Fähigkeiten oder Kompetenzen zertifizieren.
Sie werden bewertet, qualitätsgesichert und können einzeln genutzt oder zu größeren Qualifikationen kombiniert („gestapelt“) werden.

Erweiterte Definition
MC als Enablerer für Abschlüsse

Micro-Credentials als Unterstützung für Studierende auf dem Weg zum **Abschluss**

Micro-Credentials ermöglichen es Studierenden, ihren Bildungsweg in **flexiblen**, modularen Schritten zu gestalten.

Da sie klar abgegrenzte Kompetenzen mit definierten Lernzielen, Workload und Prüfungsformaten **zertifizieren**, können Lernende in ihrem eigenen Tempo vorankommen, gezielt **Kompetenzlücken** schließen oder über nicht-traditionelle Wege in ein Studium einsteigen.

Sind Micro-Credentials stapelbar und institutionell **anerkannt**, dienen sie als Bausteine, die sich zu größeren Abschlüssen wie Zertifikaten, Diplomen oder vollständigen Studienabschlüssen kombinieren lassen.

So wird Hochschulbildung **zugänglicher, inklusiver** und anpassungsfähiger – ohne dass die **wissenschaftliche Qualität** verloren geht.

Europass?

Europass speichert nicht die Daten aller Menschen an einem zentralen Ort.

Stattdessen gilt:

- **Es bietet eine persönliche digitale Wallet**, in der jede Person selbst kontrolliert, welche Daten sie hochlädt und welche sie teilt.
- **Es definiert gemeinsame Standards** (Metadaten, Formate, verifizierbare Nachweise), damit Hochschulen, Arbeitgeber und Lernende Kompetenzen und Qualifikationen europaweit interoperabel austauschen können.
- **Es unterstützt Mobilität**, weil Europass-Dokumente (z. B. Lebenslauf, Diploma Supplement, Micro-Credentials) überall dem gleichen europäischen Standard folgen – unabhängig davon, welche Institution sie ausstellt.

Rein theoretisch könnte ich mir hier eigenhändig einen Bachelor Europaweit zusammenstecken über Jahre hinweg….

ESCO European classification of Skills, Competences, Occupations and Qualifications

Ein **ESCO Skill** ist eine **standardisierte, europäisch harmonisierte Beschreibung einer Fähigkeit**, z. B.:

- *"apply object-oriented programming"*
- *"design relational databases"*
- *"use version control systems"*
- *"conduct user requirements analysis"*
- *"apply copyright and data protection standards"*

Diese Skills haben jeweils:

- eine **eindeutige ID**
- eine **Standarddefinition**
- Verlinkungen zu:
  - **Berufen (Occupations)**
  - **Aufgaben (Tasks)**
  - **anderen Skills**

Vorteil

- Nähe Hochschule ⇔ Industrie
- Internationalisierung / automatische Übersetzung

- **Skills (Fähigkeiten)**
- **Competences (Kompetenzen)**
- **Qualifications (Abschlüsse)**
- **Occupations (Berufe)**

Gründe gegen Microcredentials

- 5 ECTS ist unsere minimale Akkreditierungsgröße in BW
- Man kann ein vorsichtig abgestimmtes Curriculum nicht zersplittern
- Accreditierung von MC ist ein Problem
- Wer sind die Kunden

Gründe gegen ESCO Standards

Zwar …

- Industrie relevant
- Skill Profile
- Abstimmung zwischen Modulen
- Lernangebote passen sich an den Bedarf an
- Automatische Übersetzung in 65 Sprachen

Aber …

- Modulbeschreibungen anpassen
- Statische Curricula (8 Jahre)
- Akkreditierung langwieriger Prozess

Gründe gegen Europass

- Handarbeit
- Was ist Europass
- Ich habe kein Siegel
- Ich benötige das nicht

Use Case – ein Grund dafür!

- Internationale Studierende in Baden Württemberg
- Studiengebühren: 1500€ / Semester für nicht Europäer
- Schwierigkeiten mit deutscher Prüfungskultur
- Sprachhürde
- Hohes Risiko für Studierende
- Hohes Risiko für Duale Partner der DHBW
- Fachkräftemangel

Lösungsansatz

⇒ Anrechenbare Zeritifikate / Microcredentials

⇒ Hybride Vorlesungen

Informatik Curriculum (Beispiel)

Semester 1	ECTS-LP	SWS	PL (B,U)*	Semester 2	ECTS-LP	SWS	PL (B,U)*	Semester 3	ECTS-LP	SWS	PL (B,U)*	Semester 4	ECTS-LP	SWS	PL (B,U)*	Semester 5	ECTS-LP	SWS	PL (B,U)*	Semester 6	ECTS-LP	SWS	PL (B,U)*
Kernmodule																							
Mathematik I	5	5	B	Mathematik II	5	5	B	Mathematik III		3	B	Mathematik III	6	3	B								
Theoretische Informatik I	5	5	B	Theoretische Informatik II	5	5	B	Theoretische Informatik III		4	B	Theoretische Informatik III	6	2	B								
Programmieren		3		Programmieren	5	4	B	Software Engineering I		4	B	Software Engineering I	9	4	B								
Schlüsselqualifikationen		3	B	Schlüsselqualifikationen	5	3		Datenbanksysteme		3	B	Datenbanksysteme	6	3	B								
				Technische Informatik I	5	5	B	Technische Informatik II		6	B	Technische Informatik II	8	2	B								
												IT-Sicherheit	5	5	B	Studienarbeit		1		Studienarbeit	10	1	B
Praxisprojekt I					20	1	U,U	Praxisprojekt II					20	1	U,B,B	Praxisprojekt III	8	1	U+U	Bachelorarbeit	12	1	B
Studienrichtungsmodule																							
Informatik																							
Scientific Programming Lab	5	5	B	Grundlagen Intelligente Systeme	5	5	B									Neue Konzepte der Informatik	5	5	B	Technologie Seminar Angewandte Künstliche Intelligenz	5	5	B
Web Engineering		3		Web Engineering	5	3	B									Software Engineering II	5	5	B				
Wahlmodule																							
								Data Science 2601	5	6	B		5	3	B	Sprach und Bild Verarbeitung	5	5	B	International Project	5	5	B
								Design Thinking T4INF9006		2		Design Thinking T4INF9006	5	3	B	Gamification	5	5	B	Games	5	5	B
																				Informatik, Unternehmen und Gesellschaft	5	5	B

Ein Curriculum aus potentiellen Microcredentials

Informatik Curriculum (Beispiel)

Kernmodule

Semester 1

- Mathematik I (5 ECTS)
- Theoretische Informatik I (5 ECTS)
- Programmieren (5 ECTS)
- Schlüsselqualifikationen (3 ECTS)

Semester 2

- Mathematik II (5)
- Theoretische Informatik II (5)
- Programmieren (4)
- Schlüsselqualifikationen (3)
- Technische Informatik I (5)

Wahlfächer

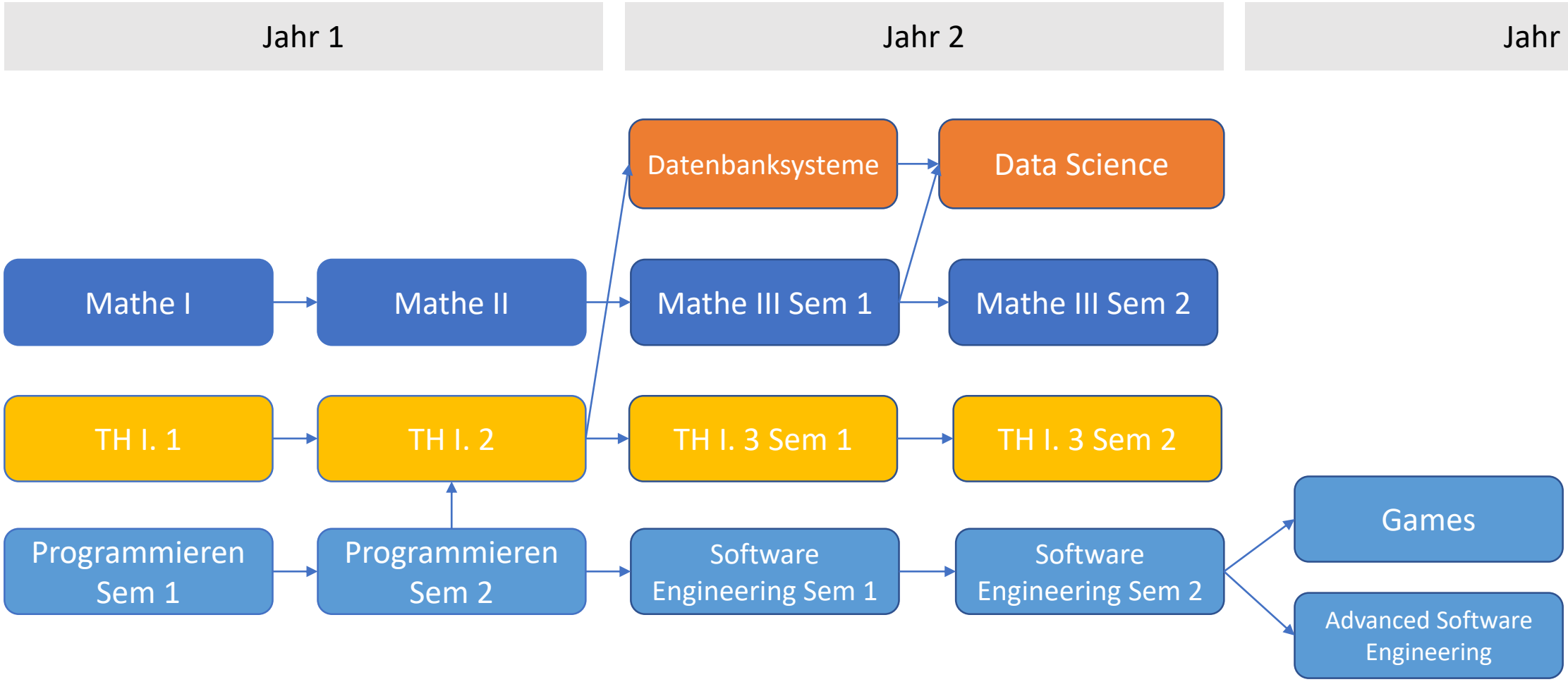
Früh (Semester 1–2)

- Scientific Programming Lab
- Grundlagen Intelligente Systeme
- Web Engineering

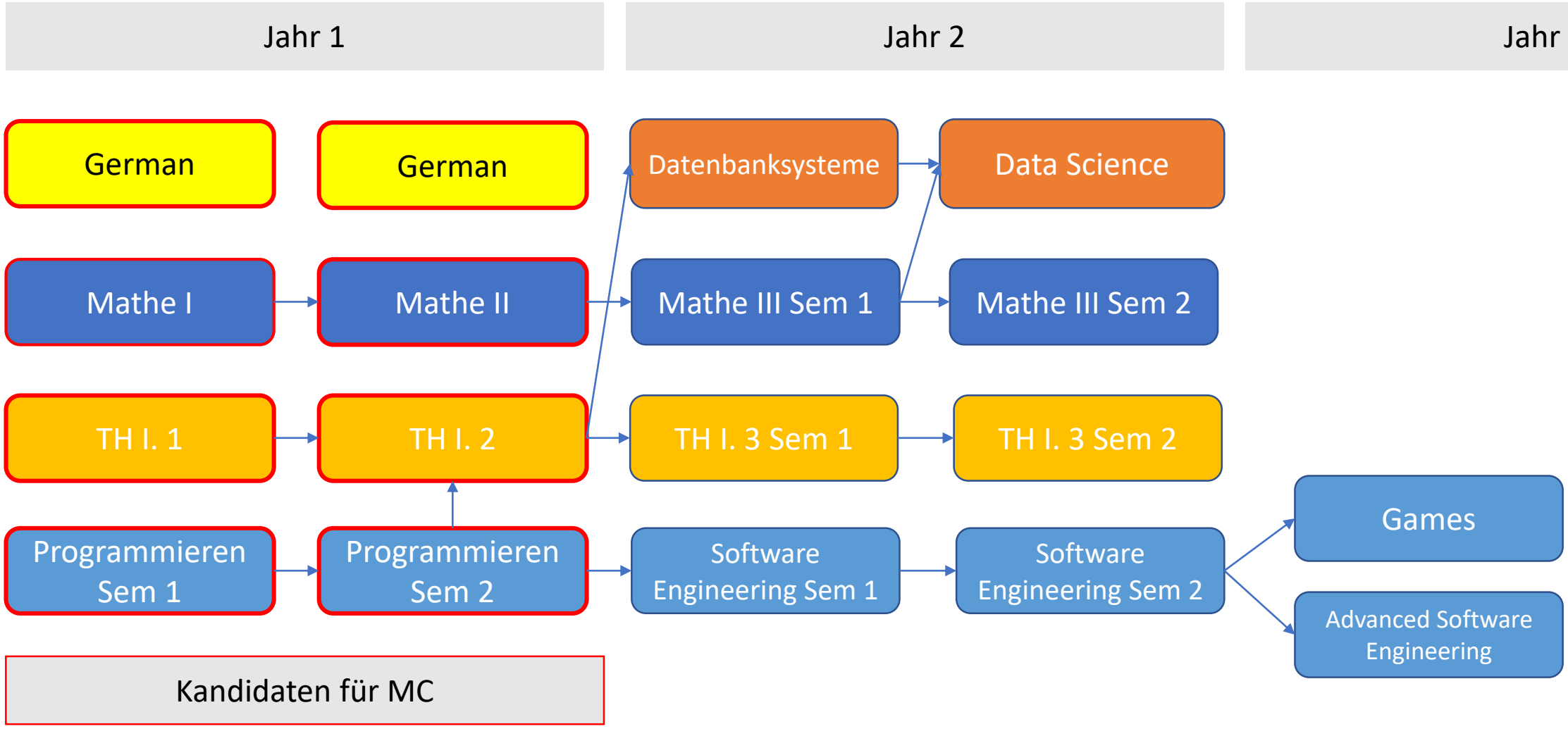
Später (Semester 5–6)

- Neue Konzepte der Informatik
- Software Engineering II
- Technologie Seminar / Angewandte Künstliche Intelligenz
- Sprach- und Bildverarbeitung
- Gamification
- International Project
- Games
- Informatik, Unternehmen und Gesellschaft

Lernpfade im Curriculum (Beispielhaft)



MC zerstören keine Curricula



PROGRAMMIEREN (T4INF1004)

DHBW
Duale Hochschule

✅ PROGRAMMIEREN (T4INF1004)

Semester: 1–2
ECTS: 7
Workload: 210 h
Präsenz: 96 h
Selbststudium: 114 h
Lehrformen: Vorlesung, Übung, Labor
Prüfungsform: Entwurf (Projekt / praktische Leistung)
Sprache: Deutsch

WORKLOAD UND ECTS-LEISTUNGSPUNKTE

WORKLOAD INSGESAMT (IN H)	DAVON PRÄSENZZEIT (IN H)	DAVON SELBSTSTUDIUM (IN H)	ECTS-LEISTUNGSPUNKTE
210	96	114	7

PDF

QUALIFIKATIONSZIELE UND KOMPETENZEN

FACHKOMPETENZ

Die Studierenden kennen die Grundelemente der prozeduralen und der objektorientierten Programmierung. Sie können die Syntax und Semantik dieser Sprachen. Sie können die für eine Problemstellung passenden Datenstrukturen auswählen und anwenden.

METHODENKOMPETENZ

Die Studierenden können ein Programmdesign selbstständig entwerfen, codieren und ihr Programm auf Funktionsfähigkeit testen. Sie setzen verschiedene Strukturierungsmöglichkeiten ein um einfache Programme selbstständig zu erstellen, und sie können die für eine Problemstellung passenden Datenstrukturen auch implementieren. Die Studierenden können eine Entwicklungsumgebung verwenden um Programme zu erstellen, zu strukturieren und auf Fehler hin zu untersuchen (inkl. Debugger).

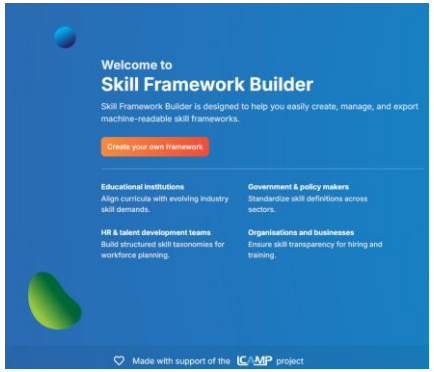
PERSONALE UND SOZIALE KOMPETENZ

Die Studierenden können ihren Programmentwurf sowie dessen Codierung im Team diskutieren, begründen und entwickeln. Sie können existierenden Code analysieren und beurteilen. Sie können sich selbstständig in Entwicklungsumgebungen einarbeiten und diese zur Programmierung und Fehlerbehebung einsetzen.

ÜBERGREIFENDE HANDLUNGSKOMPETENZ

Die Studierenden können eigenständig einfache Problemstellungen der Praxis analysieren und zu deren Lösung Programme entwerfen, programmieren und testen.

https://skilldata.info/



Knowledge
INNOVATION
Centre

Backend Systeme der DACH Hochschulen sind ein Thema für sich – Show Stopper

PROGRAMMIEREN (T4INF1004)

✅ PROGRAMMIEREN (T4INF1004)

Semester: 1–2

ECTS: 7

Workload: 210 h

Präsenz: 96 h

Selbststudium: 114 h

Lehrformen: Vorlesung, Übung, Labor

Prüfungsform: Entwurf (Projekt / praktische Leistung)

Sprache: Deutsch

📚 Inhalte des Moduls PROGRAMMIEREN

(Ein einziges großes Modul über 2 Semester)

1) Prozedurale Programmierung

(Grundlagen im 1. Semester)

- Algorithmenbeschreibung
- Datentypen
- Ein-/Ausgabe und Dateiverarbeitung
- Operatoren
- Kontrollstrukturen
- Funktionen
- Stringverarbeitung
- Strukturierte Datentypen
- dynamische Datentypen
- Zeiger
- Speicherverwaltung

2) Objektorientierte Programmierung

(Vertiefung im 2. Semester)

- objektorientierter Programmentwurf
- Klassenkonzept
- Operatoren
- Überladen von Operatoren, Methoden
- Vererbung & Überschreiben
- Polymorphismus
- Templates/Generics
- Klassenbibliotheken
- Speicherverwaltung & Garbage Collection

🎯 Qualifikationsziele & Kompetenzen

Fachkompetenz

Studierende …

- kennen die **Grundelemente der prozeduralen und objektorientierten Programmierung.**
- verstehen **Syntax und Semantik** der verwendeten Sprachen.
- können **geeignete Datenstrukturen auswählen und anwenden.**
- können **Programmentwürfe erstellen**, Programme codieren und testen.

Methodenkompetenz

- Selbstständiger Programmentwurf
- Fehleranalyse mit Debuggern
- Nutzung von Entwicklungsumgebungen
- Strukturierung von Code und Anwendungen

Personale & soziale Kompetenz

- Teamarbeit bei Entwurf & Codierung
- Code-Analyse und gemeinsames Review
- eigenständige Einarbeitung in IDEs

Übergreifende Handlungskompetenz

- Fähigkeit, **einfache Problemstellungen** selbst zu analysieren
- Programme **eigenständig entwerfen, programmieren und testen**

1. Fachkompetenz → ESCO Hard Skills

Deine Kompetenz	Passender ESCO Skill
Grundelemente der prozeduralen & OOP kennen	"apply programming principles", "object-oriented programming", "procedural programming"
Syntax & Semantik verstehen	"interpret programming language syntax and semantics"
Datenstrukturen auswählen & anwenden	"use data structures", "analyse algorithms"
Programme entwerfen, codieren und testen	"develop software", "write code", "test software", "debug software"

programming object oriented apply

Search

English

Copy 3 selected

Copy skill names

Copy skill descriptions

Use object-oriented programming

Utilise specialised ict tools for programming paradigm based on the concept of objects, which can contain data in the form of fields and code in...

Object-oriented modelling

The object-oriented paradigm, which is based on classes, objects, methods and interfaces and their application in software design and analysis,...

Objective-c

The techniques and principles of software development, such as analysis, algorithms, coding, testing and compiling of programming paradigms ...

Apply basic programming skills

List simple instructions for a computing system to solve problems or perform tasks at a basic level and with appropriate guidance where needed.

Use automatic programming

Utilise specialised software tools to generate computer code from specifications, such as diagrams, structured informations or other means...

Computer programming

The techniques and principles of software development, such as analysis, algorithms, coding, testing and compiling of programming paradigms...

Grundelemente der prozeduralen & OOP kennen

Use object-oriented programming

Utilise specialised ict tools for programming paradigm based on the concept of objects, which can contain data in the form of fields and code in the form of procedures. use programming languages which support this method such as java and c++.

Apply basic programming skills

List simple instructions for a computing system to solve problems or perform tasks at a basic level and with appropriate guidance where needed.

Computer programming

The techniques and principles of software development, such as analysis, algorithms, coding, testing and compiling of programming paradigms (e.g. object oriented programming, functional programming) and of programming languages.

Search skills

Search ...

Find

Show filters

- K - knowledge
  - 00 - generic programmes and qualifications
  - 01 - education
  - 02 - arts and humanities
  - 03 - social sciences, journalism and information
  - 04 - business, administration and law
  - 05 - natural sciences, mathematics and statistics
  - 06 - information and communication technologies (icts)
    - 061 - information and communication technologies (icts)
      - 0610 - information and communication technologies (icts) not further defined
      - 0611 - computer use
      - 0612 - database and network design and

computer programming

Download

knowledge > information and communication technologies (icts) > information and communication technologies (icts) > software and applications development and analysis > computer programming

skills > working with computers > using digital tools for collaboration, content creation and problem solving > using digital tools for collaboration and productivity > digital content creation > computer programming

Concept overview

Description

The techniques and principles of software development, such as analysis, algorithms, coding, testing and compiling of programming paradigms (e.g. object oriented programming, functional programming) and of programming languages.

Alternative Labels

AngularJS | Caml | Fantom | computer programming | Subversion

Seed7 | STL | Fortran | IDL | OCaml | COMAL | Visual Basic .NET

1. Fachkompetenz → ESCO Hard Skills

Deine Kompetenz	Passender ESCO Skill
Grundelemente der prozeduralen & OOP kennen	"apply programming principles", "object-oriented programming", "procedural programming"
Syntax & Semantik verstehen	"interpret programming language syntax and semantics"
Datenstrukturen auswählen & anwenden	"use data structures", "analyse algorithms"
Programme entwerfen, codieren und testen	"develop software", "write code", "test software", "debug software"

Framework Builder | Skill Finder

syntax and semantic

Search

English

Copy 3 selected

Copy skill names

Copy skill descriptions

Grammar

The set of structural rules governing the composition of clauses, phrases, and words in any given natural language.

Semantics

The branch of linguistics that studies meaning; it analyses of words, phrases, signs, and symbols and the relation between them.

Create semantic trees

Create coherent lists and hierarchies of concepts and terms to ensure consistent indexing in knowledge organisation systems.

Theoretical lexicography

The academic field dealing with the syntagmatic, paradigmatic, and semantic relationships within the vocabulary of a certain language.

Sass

The computer language sass is a style sheet language that conveys the presentation of structured documents. these documents have to...

Manage ict semantic integration

Oversee integration of public or internal databases and other data, by using semantic technologies to produce structured semantic output.

✅ Generated ESCO-style Skill: “interpret programming language syntax and semantics”

Skill name (preferred term)

Interpret programming language syntax and semantics

Alternative labels

- Understand programming language structure
- Analyse syntax and semantics of programming languages
- Interpret formal language constructs
- Apply programming language rules

Skill type

Skill/competence – Knowledge & Cognitive skill
(compatible with ESCO’s Digital Competence → Software development → Programming)

Skill definition (ESCO format)

Interpret and apply the formal rules that define the structure (syntax) and meaning (semantics) of a programming language in order to understand, analyse, and correctly execute or reason about code. This includes recognising valid constructs, detecting invalid expressions, and understanding how statements are evaluated by the language model.

Knowledge components

This skill requires knowledge of:

- **Syntax rules** (grammars, operators, precedence, scoping, typing rules)
- **Semantics** (operational, declarative, or denotational meaning of constructs)
- **Control flow and evaluation order**
- **Type systems and data models**
- **Error and exception behavior**

Skill components / performance criteria

A person with this skill can:

- Read and interpret code according to formal grammar and language specifications
- Identify syntactically incorrect statements
- Explain how an expression or statement will be evaluated at runtime
- Identify semantic errors (e.g., misuse of types, undefined behavior)
- Predict program outcomes based on the language’s execution model
- Map code constructs to abstract syntax concepts
- Understand documentation, syntax diagrams, and formal specifications

---

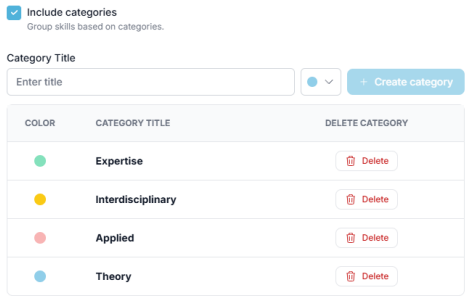
Context of use (ESCO-style examples)

- Reviewing or debugging code
- Learning or switching to a new programming language
- Designing or analysing algorithms
- Working with compiler/interpreter outputs
- Creating teaching materials or assessing code submissions

Build a new framework

Follow this step by step guide to create a new framework.

- Basic setup — Step 1
- Additional options — Step 2
- Define proficiency levels — Step 3
- Confirm and create — Step 4



Cancel

Assign a classification of "skill", "knowledge" or "language skill" to every skill you add.

Include proficiency levels
Let the skills in your framework have a proficiency level. Otherwise the skills are binary.

Number of proficiency levels

2 3 4 5 6 7 8 9 10

Include categories
Group skills based on categories.

Category Title

Interdisciplinary

+ Create category

COLOR	CATEGORY TITLE	DELETE CATEGORY
	Applied	Delete
	Theory	Delete

Fields marked with* are required

Go back

Continue

Computer Science DHBW

v1 | Draft Manage + Add skills

About | Skills (0) | Qualifications (0) | Job profiles (0)

Add individual skills | Bulk import skills

Clear Import from a file

Skill title *

Interpret programming language syntax and semantics Customise

Skill description

Interpret and apply the formal rules that define the structure (syntax) and meaning (semantics) of a programming language in order to understand, analyse, and correctly execute or reason about code. This includes recognising valid constructs, detecting invalid expressions, and understanding how statements are evaluated by the language model.

Context of use:
Reviewing or debugging code
Learning or switching to a new programming language

Customise

Related skills

Refetch related skills

- Grammar — Knowledge
- Semantics — Knowledge
- Sas Language — Knowledge
- Web Programming — Knowledge
- Computer Science — Knowledge
- Computer Programming — Knowledge

View more

Category *

Theory

Manage categories

ESCO classification *

Skill/Competence

Customise

Define levels *

Uncheck specific levels if you would like to make those levels unavaliable for that particular skill.

Level 1: First exposure

Level 2: Depth

Level description

Explain how an expression or statement will be evaluated at runtime
Identify semantic errors (e.g., misuse of types, undefined behavior)
Predict program outcomes based on the language's execution model
Map code constructs to abstract syntax concepts
Understand documentation, syntax diagrams, and formal specifications

Level 3: Expert

Level 1: First exposure

Level description

Syntax rules (grammars, operators, precedence, scoping, typing rules)
Semantics (operational, declarative, or denotational meaning of constructs)
Control flow and evaluation order
Type systems and data models
Error and exception behavior

Fields marked with * are required

+ Add skills to framework

Computer Science DHBW

v1 | Draft Manage Add skills

About Skills (1) Qualifications (0) Job profiles (0)

List Sunburst chart Add skills

Skills Dashboard

Track your progress

Categories
1

Skills
1

Search categories...

Categories Overview

Theory
1 skills

Skills Visualization

Interactive overview

Framework Builder | Skill Finder

"use data structures", "analyse algorithms"

Search

English

- English
- Arabic
- Bulgarian
- Czech
- Danish
- German
- Greek

Copy 0 selected | Copy skill names | Copy skill descriptions

Use data processing techniques

Gather, process and analyse relevant data and information, properly store and update data and represent figures and data using charts and...

Apply statistical analysis techniques

Use models (descriptive or inferential statistics) and techniques (data mining or machine learning) for statistical analysis and ict tools to analyse data,...

Use specific dat software

Use specific software for data statistics, spreadsheets, and d possibilities in order to make reports to managers,...

Analyse test data

Interpret and analyse data collected during testing in order to formulate conclusions, new insights or solutions.

Establish data processes

Use ict tools to apply mathematical, algorithmic or other data manipulation processes in order to create information.

Analyse data about clients

Study data about clients, visitors, customers or guests. gather, process and analyse data about their characteristics, needs and buying behaviours.

Use databases

Use software tools for managing and organising

Think analytically

Produce thoughts using logic and reasoning in

Analyse scientific data

Collect and analyse scientific data resulting from

1. Fachkompetenz → ESCO Hard Skills

Deine Kompetenz	Passender ESCO Skill
Grundelemente der prozeduralen & OOP kennen	"apply programming principles", "object-oriented programming", "procedural programming"
Syntax & Semantik verstehen	"interpret programming language syntax and semantics"
Datenstrukturen auswählen & anwenden	"use data structures", "analyse algorithms"
Programme entwerfen, codieren und testen	"develop software", "write code", "test software", "debug software"

Nichts Neues…

FACHKOMPETENZ

Die Studierenden kennen die Grundelemente der prozeduralen und der objektorientierten Programmierung. Sie können die Syntax und Semantik dieser Sprachen. Sie können die für eine Problemstellung passenden Datenstrukturen auswählen und anwenden.

Data models

The techniques and existing systems used for structuring data elements and showing relationships between them, as well as methods for interpreting the data structures and relationships.

Algorithms

The self-contained step-by-step sets of operations that carry out calculations, data processing and automated reasoning, usually to solve problems.

Use object-oriented programming

Utilise specialised ict tools for programming paradigm based on the concept of objects, which can contain data in the form of fields and code in the form of procedures. use programming languages which support this method such as java and c++.

Apply basic programming skills

List simple instructions for a computing system to solve problems or perform tasks at a basic level and with appropriate guidance where needed.

Computer programming

The techniques and principles of software development, such as analysis, algorithms, coding, testing and compiling of programming paradigms (e.g. object oriented programming, functional programming) and of programming languages.

Deine Kompetenz
Grundelemente der prozeduralen & OOP kennen
Syntax & Semantik verstehen
Datenstrukturen auswählen & anwenden
Programme entwerfen, codieren und testen

Interpret Programming Languages Syntax and Semantics (Level 1) : Interpret and apply the formal rules that define the structure (syntax) and meaning (semantics) of a programming language in order to understand, analyse, and correctly execute or reason about code. This includes recognising valid constructs, detecting invalid expressions, and understanding how statements are evaluated by the language model. Context of use: Reviewing or debugging code Learning or switching to a new programming language Designing or analysing algorithms Working with compiler/interpreter outputs Creating teaching materials or assessing code submissions

2. Methodenkompetenz → ESCO Cognitive / Technical Skills

Deine Kompetenz	Passender ESCO Skill
Selbstständiger Programmentwurf	"design software architecture", "apply software design patterns"
Fehleranalyse mit Debuggern	"debug software", "use debugging tools"
Nutzung von Entwicklungsumgebungen	"use integrated development environment (IDE)"
Strukturierung von Code & Anwendungen	"apply coding standards", "produce clean code"

Methodenkompetentz (5 Min)

Selbstständig arbeiten
Entwickeln eigener arbeitsweisen aus eigenem antrieb, mit wenig oder gar keiner aufsicht, und eigenständiges bewältigen der arbeiten.
Ikt-debugging-werkzeuge
Die zum testen und zur fehlersuche von programmen und quellcode verwendeten ikt-werkzeuge wie gnu debugger (gdb), intel debugger (idb), microsoft visual studio debugger, valgrind und windbg.
Softwareprobleme identifizieren und beheben
Durch analyse von testergebnissen computercodes reparieren. zu diesem zweck fehler finden, die die software dazu veranlassen, ein falsches oder unerwartetes resultat auszugeben, und diese fehler beheben.
Rechnergestützte softwareentwicklungswerkzeuge verwenden
Softwaretools (case) zur unterstützung des entwicklungszyklus, der entwicklung und implementierung von software und hochwertigen anwendungen, die leicht betrieben werden können, benutzen.
Softwaredesign entwickeln
Umsetzung diverser anforderungen in ein klares, strukturiertes softwaredesign.

3. Personale & soziale Kompetenz → ESCO Soft Skills

Deine Kompetenz	Passender ESCO Skill
Teamarbeit bei Entwurf & Codierung	"work in a software development team", "collaborate in multidisciplinary teams"
Code-Analyse & gemeinsames Review	"perform code review", "evaluate software quality"
eigenständige Einarbeitung in IDEs	"self-directed learning", "adapt to new technologies"

Personale Kompetenzen (2 Min)

In teams arbeiten

Vertrauensvolle zusammenarbeit in einem team, in dem jedes teammitglied seinen aufgabenbereich hat.

Iuk-code-reviews durchführen

Systematisches untersuchen und überprüfen von rechnergestütztem quellcode zur ermittlung von fehlern in allen phasen der entwicklung und zur verbesserung der allgemeinen softwarequalität.

Selbstständig arbeiten

Entwickeln eigener arbeitsweisen aus eigenem antrieb, mit wenig oder gar keiner aufsicht, und eigenständiges bewältigen der arbeiten.

4. Übergreifende Handlungskompetenz → ESCO Transversal Skills

Deine Kompetenz	Passender ESCO Skill
einfache Problemstellungen analysieren	"analyse technical requirements", "identify user needs"
Programme eigenständig entwerfen, programmieren & testen	"develop software", "implement solutions", "apply problem-solving skills", "test software"

Übergreifende Handlungskompetenz (2 Min)

Probleme lösen

Lösungen für praktische, betriebsbezogene oder konzeptionelle probleme in einer vielzahl unterschiedlicher kontexte finden.

ESCO Was soll‘s? (2 h / Modul)

Fachkompetenz
- Data models
- Algorithms
- Use object-oriented programming
- Apply basic programming skills
- Computer programming
- Interpret programming language syntax and semantics *(Ergänzt)*

Methodenkompetenz
- ICT debugging tools
- Identify and resolve software problems
- Use computer-aided software engineering tools (CASE tools)
- Develop software design

Personale & soziale Kompetenzen
- Work independently
- Work in a team

Übergreifende Handlungskompetenz
- Perform code reviews
- Problem solving

QUALIFIKATIONSZIELE UND KOMPETENZEN

FACHKOMPETENZ

Die Studierenden kennen die Grundelemente der prozeduralen und der objektorientierten Programmierung. Sie können die Syntax und Semantik dieser Sprachen. Sie können die für eine Problemstellung passenden Datenstrukturen auswählen und anwenden.

METHODENKOMPETENZ

Die Studierenden können ein Programmdesign selbstständig entwerfen, codieren und ihr Programm auf Funktionsfähigkeit testen. Sie setzen verschiedene Strukturierungsmöglichkeiten ein um einfache Programme selbstständig zu erstellen, und sie können die für eine Problemstellung passenden Datenstrukturen auch implementieren. Die Studierenden können eine Entwicklungsumgebung verwenden um Programme zu erstellen, zu strukturieren und auf Fehler hin zu untersuchen (inkl. Debugger).

PERSONALE UND SOZIALE KOMPETENZ

Die Studierenden können ihren Programmentwurf sowie dessen Codierung im Team diskutieren, begründen und entwickeln. Sie können existierenden Code analysieren und beurteilen. Sie können sich selbstständig in Entwicklungsumgebungen einarbeiten und diese zur Programmierung und Fehlerbehebung einsetzen.

ÜBERGREIFENDE HANDLUNGSKOMPETENZ

Die Studierenden können eigenständig einfache Problemstellungen der Praxis analysieren und zu deren Lösung Programme entwerfen, programmieren und testen.

- ✓ Redundanzen entfernt
- ✓ Überlappungen getrennt
- ✓ ESCO-Definitionen präzise zugeordnet
- ✓ Ergänzte Skill-Definition (*Syntax & Semantik verstehen*)
- ✓ 4-Ebenen-Modell (Fach, Methode, Personal, Übergreifend) durch ESCO sauber operationalisiert

Kompetenzen die nicht im Handbuch stehen

Aber den Dozierenden wichtig sind

Personale Kompetenzen (Selbstkompetenzen)

- Selbstmanagement
- Selbstwirksamkeit & Selbstvertrauen
- Reflexionskompetenz
- Resilienz

Soziale Kompetenzen

- Kommunikation
- Kooperation & Teamarbeit
- Konfliktkompetenz
- Empathie & Inklusion

Das lernende Curriculum => ein anderes Thema

Games and Gaming (Microcredential)

5 ECTS | Seminar | Project | Prototype & Presentation | Microcredential

Workload	Presence Time	Self-Study	Responsible
110 hours total	50 hours	60 hours	Kay Berkling

📘 Description

This course integrates knowledge from various disciplines—including Gamification, (Advanced) Software Engineering, Programming, Web Engineering, Databases or Communication & Networks—applying them to game development projects using current technologies. Students will work with game engines to implement real prototypes, addictive patterns, user analytics, architecture and user experience.

Key Focus Areas:

- **Technology Specialization:** Teams focus on specific platforms (e.g. Unity, Unreal Engine, etc.) and share expertise with peers.
- **Game Design Patterns:** Psychological aspects and patterns of games; Technical implementations of games.
- **Graphics & Animation:** as desired
- **Personal Growth:** Development of self-directed work habits and group dynamics.

Important Note: Due to the advanced prerequisites and emphasis on self-regulated learning, this course is not recommended for students unwilling to invest significant time or actively contribute to shaping the course quality. Learning objectives are collaboratively defined by participants at the start.

🎯 Skills & Competencies

Fachkompetenz (Subject Competence)

- Unreal engine
- Unity (digital game creation systems)
- C#
- C++
- Computer programming
- Integrated development environment software
- Tools for software configuration management

Methodenkompetenz (Methodological Competence)

- Align software with system architectures
- Perform data analysis
- Debug software
- Apply gaming psychology
- Design user interface
- Create prototype of user experience solutions

Personale und Soziale Kompetenz (Personal & Social Competence)

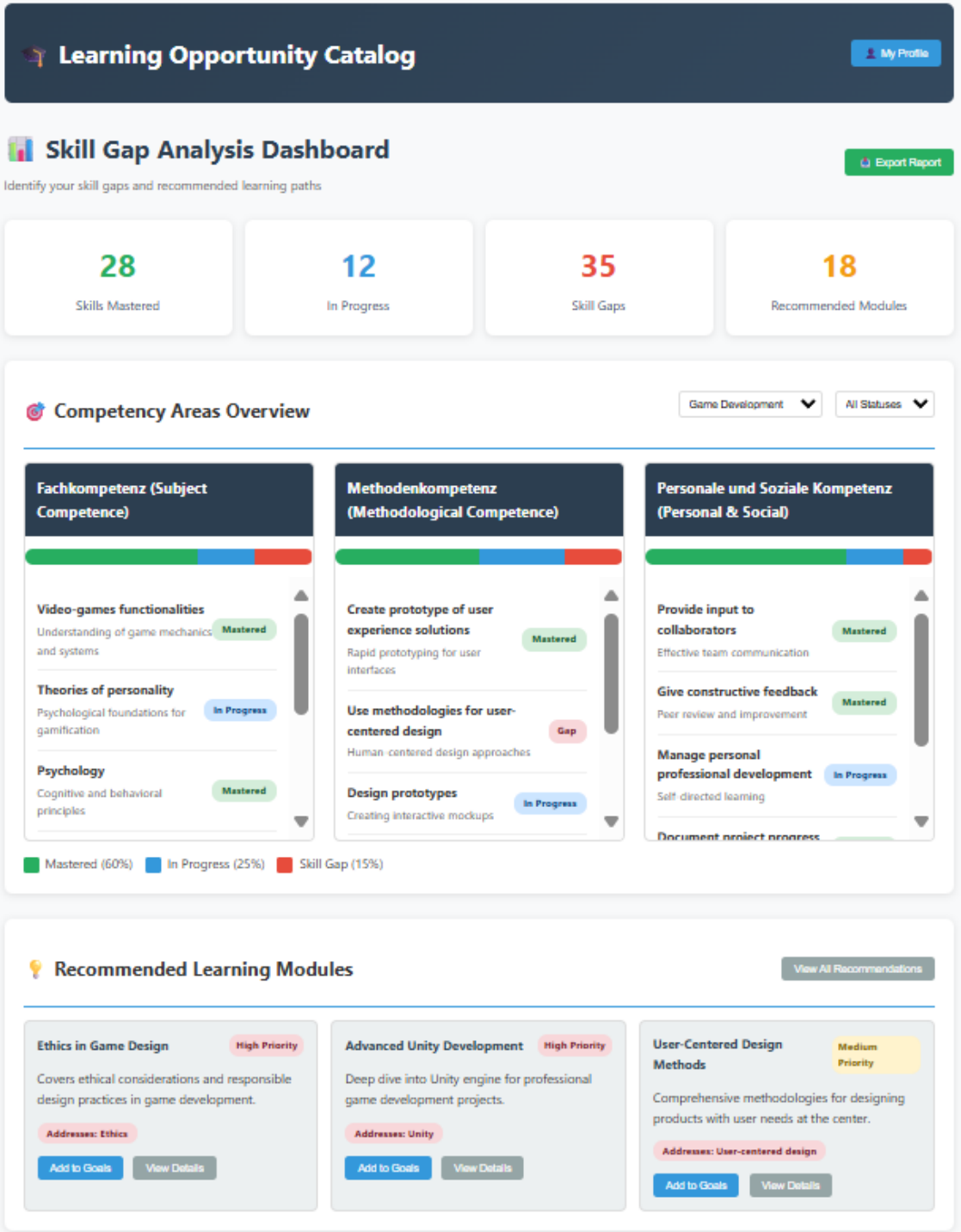
- Use learning strategies
- Critically reflect on artistic production processes
- Exercise self-reflection
- Work independently
- Assume responsibility
- Manage teamwork

MC Katalog

- Personalisierte Wege
- Personalisierte Vorschläge
- Skill Profile
- Ziel Definition
- Übersicht
- Suchbar

ESCO Standards sind die Grundlage für:

- **Microcredentials**
- **Selbstgesteuerte Lernpfade**
- **KI-basierte Empfehlungen für Lernende**
- **Vergleichbarkeit der Module in Europa**
- **Automatisierte Curriculumsanalyse**
- **Skill-Transparenz zwischen Work & Study (Dualität)**



🎯 Competency Areas Overview

Fachkompetenz (Subject Competence)

Computer programming
Programmieren I, II, Software Engineering I
Mastered

C#
Games and Gaming, Software Engineering II
Mastered

C++
Games and Gaming, Eingebettete Systeme
Mastered

Video-games functionalities
Gamification
Mastered

Psychology
Mastered

Methodenkompetenz (Methodological)

Create prototype of user experience solutions
Gamification, Games and Gaming
Mastered

Use methodologies for user-centered design
Gamification, Web-Engineering II
Mastered

Design prototypes
Gamification, Software Engineering I
Mastered

Apply gaming psychology
Gamification, Games and Gaming
Mastered

Design user interface
Games and Gaming, Web-
In Progress

Personale und Soziale Kompetenz

Provide input to collaborators to guide their work
Gamification, Games and Gaming
Mastered

Give constructive feedback
Gamification, Games and Gaming
Mastered

Manage personal professional development
Gamification, Games and Gaming
Mastered

Document project progress
Gamification, Games and Gaming
Mastered

Work independently
Games and Gaming,
In Progress

Mastered (75%) In Progress (15%) Skill Gap (10%)

My Skills Journey

How your skills evolved across courses and over time.

Student: Alex Müller Program: BSc Computer Science (Modular Track) Period: WS 2024 – WS 2025

Your Skill Profile (Aggregate)

Skills you've developed across all your courses, mapped to ESCO categories.

Software Engineering

Collaboration & Teamwork

Requirements & Analysis

Autonomy & Project Ownership

Course Timeline & Skill Contributions

Each course you've taken—and the key skills it helped you build.

ing

SS 2025

Web Development

ECTS: 5 | Prof. Z

Key skills gained:

- REST API design
- Frontend frameworks
- Responsive UI

WS 2025

Software Engineering

ECTS: 5 | Prof. X

Key skills gained:

- System architecture
- Agile team workflows
- Technical documentation
- Peer code review

Planned: SS 2026

DevOps & Cloud

ECTS: 5 | —

Expected skills:

- CI/CD pipelines
- Containerization (Docker)
- Infrastructure as Code

How Your Skills Build Over Time

Foundational skills (e.g., Git, OOP) from early courses enabled deeper work in **Software Engineering**, where you practiced system design and teamwork. Future courses like *DevOps* will rely on this base.

Git → Agile workflows | OOP → System architecture | Debugging → CI/CD testing

Kay Berkling

Aussteller

Software Engineering (Teil 1/3) Spezifikation

Projektarbeit

Software Engineering Part 1: Specification

Kommunikation (SE1 Teil 1)

Peer Reviews

Präsentieren

Bloggen

Kommunikation

Software (SE1 Teil 1)

Dokumentation

Software

Projekt Management (SE1 Teil1)

Planen

Software Engineering (Teil 1/3) Spezifikation

Verliehen von

Duale Hochschule Baden Württemberg DHBW

Angaben zur Leistung

Lernumfang
33 Stunden

EQR-Niveau
Niveau 6

Leistungspunkte
Europäisches System zur Anrechnung von Studienleistungen
3

Lernumgebungen
Formales Lernen

Lernmethode
Online

Themengebiet
Informations- und Kommunikationstechnologien (IKT)

Sprache
Englisch

Höchstdauer in Monaten
2 Monate

Art
Kurs

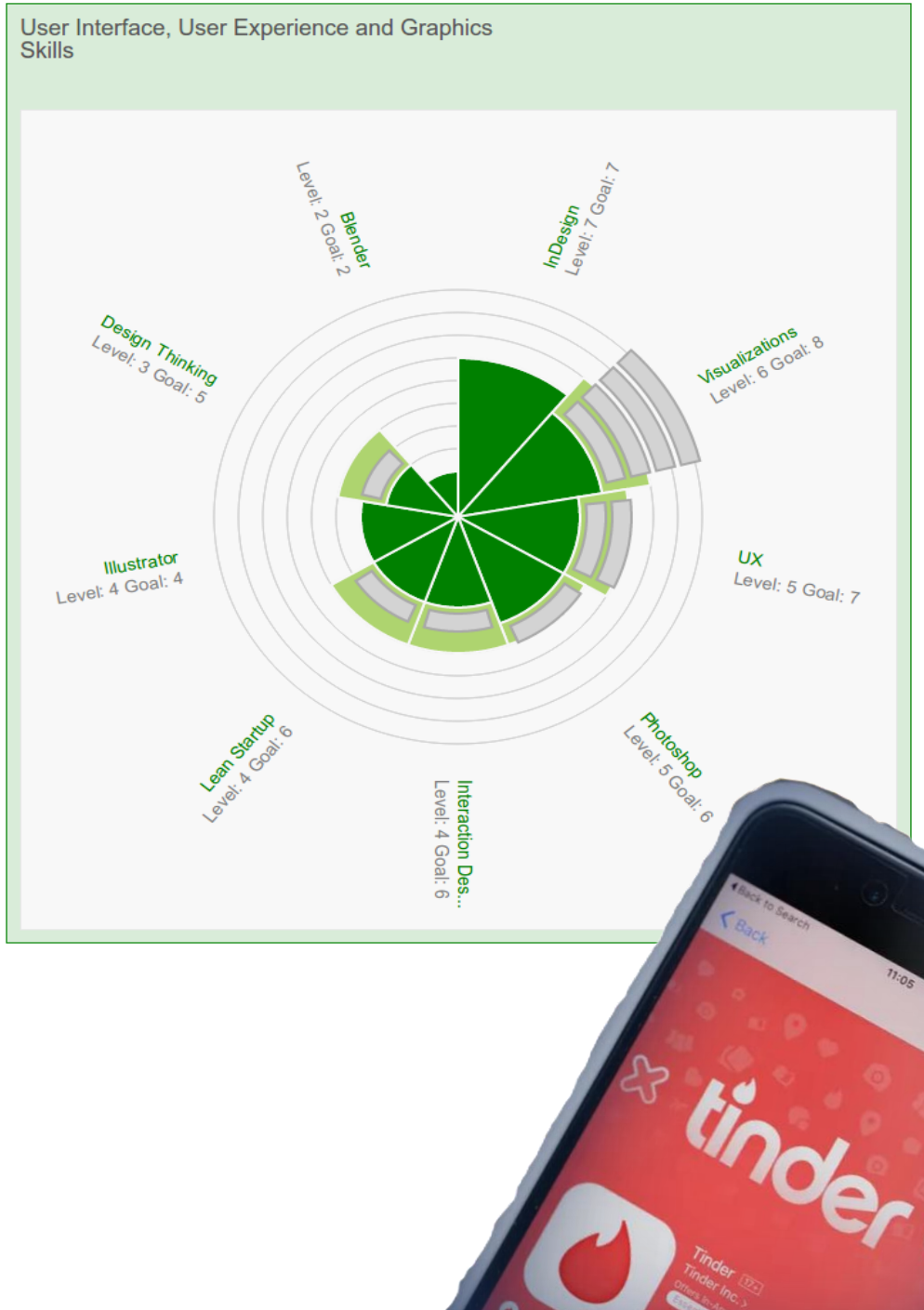
Zielgruppen
Leistungsstarke Personen

Lernaktivitäten

Projektarbeit

Europass als Vision

- **Lernende** sehen, wie sich ihr tatsächliches Kompetenzprofil über Kurse, Micro-Credentials, Praxiserfahrungen und Projekte hinweg entwickelt.
  Sie setzen gezielt Kompetenzziele und erhalten hochschulübergreifende Empfehlungen für Module, Micro-Credentials oder individuelle Lernpfade.
- **Ein europaweites Empfehlungssystem** verbindet Lernende mit passenden Studienangeboten, Praktika, Arbeitgebern und Stipendien – basierend auf **verifizierten Kompetenzen statt nur Abschlüssen**.
- **Unternehmen** finden talentierte Personen über Skills-First-Profile, können Studienverläufe mitfinanzieren, Mentoring anbieten oder direkt aus verifizierten ESCO-Skillprofilen rekrutieren.
- **Hochschulen** profitieren von einer gemeinsamen, transparenten Infrastruktur für interoperable Lernwege und ermöglichen damit echte **Durchlässigkeit, Mobilität und lebenslanges Lernen** in ganz Europa.



Universitiies
Universities
Universities
Universities
Universities
Universities
Universities
Universities
Universities
Universities
Universities
Universities
Universities
Universities
Universities
COMMUNITY
Fragen?

Build a new framework

Follow this step by step guide to create a new framework.

- Basic setup
  Step 1
- Additional options
  Step 2
- **Define proficiency levels**
  Step 3
- Confirm and create
  Step 4

Cancel

2 out of 3 levels defined

✓ Level 2: Depth

Framework title *

Depth

Level description

Applying the skill to a new environment.

Required practical experience *

1 | Experience Years

Required learning *

5 | Learning ECTS

✕ Level 3: no title

Fields marked with* are required

← Go back

Continue →

Build a new framework

Follow this step by step guide to create a new framework.

- Basic setup
  Step 1
- Additional options
  Step 2
- **Define proficiency levels**
  Step 3
- Confirm and create
  Step 4

Cancel

3 out of 3 levels defined

✓ Level 2: Depth

✓ Level 3: Expert

Framework title *

Expert

Level description

Using the skill and honing in on specific details to become an expert.

Required practical experience *

2 | Experience Years

Required learning *

5 | Learning ECTS

Fields marked with* are required

← Go back

Continue →

Build a new framework

Follow this step by step guide to create a new framework.

- Basic setup
  Step 1
- Additional options
  Step 2
- Define proficiency levels
  Step 3
- **Confirm and create**
  Step 4

These skills are specific to New Study Computer Science DHBW

Reusability type

Transversal

Use ESCO classification	Proficiency levels	Categories
Yes	**3**	**4**

PROFICIENCY LEVEL	FRAMEWORK TITLE	EXPERIENCE	KNOWLEDGE
Level 1	**First exposure**	0 Years	5 ECTS
Level 2	**Depth**	1 Years	5 ECTS
Level 3	**Expert**	2 Years	5 ECTS

CATEGORY
Expertise
Interdisciplinary
Applied
Theory

Fields marked with* are required

Cancel

← Go back

Create new framework →