

# Syllabus

## *Kursbeschreibung*

<b>Titel der Lehrveranstaltung</b>	Growth Mindset
<b>Code der Lehrveranstaltung</b>	25555
<b>Zusätzlicher Titel der Lehrveranstaltung</b>	
<b>Wissenschaftlich-disziplinärer Bereich</b>	
<b>Sprache</b>	Englisch
<b>Studiengang</b>	Master in Unternehmensführung und Innovation
<b>Andere Studiengänge (gem. Lehrveranstaltung)</b>	
<b>Dozenten/Dozentinnen</b>	dr. Silvia Sanasi, Silvia.Sanasi@unibz.it <a href="https://www.unibz.it/en/faculties/economics-management/academic-staff/person/47276">https://www.unibz.it/en/faculties/economics-management/academic-staff/person/47276</a> Dott. Cristina Maria Gangai, CristinaMaria.Gangai@unibz.it <a href="https://www.unibz.it/en/faculties/economics-management/academic-staff/person/50466">https://www.unibz.it/en/faculties/economics-management/academic-staff/person/50466</a>
<b>Wissensch. Mitarbeiter/Mitarbeiterin</b>	
<b>Semester</b>	Erstes Semester
<b>Studienjahr/e</b>	1
<b>KP</b>	14
<b>Vorlesungsstunden</b>	Module 1 – 36 hours Module 2 – 48 hours
<b>Laboratoriumsstunden</b>	Module 1 - 16 hours Module 2 - none
<b>Stunden für individuelles Studium</b>	-
<b>Vorgesehene Sprechzeiten</b>	Module 1 - 18 hours Module 2 - 24 hours

<b>Inhaltsangabe</b>	<p>M1: This project-based course introduces students to design thinking and prototyping as key tools for developing a growth mindset in the context of entrepreneurship and innovation management. Working in diverse teams, students tackle real-world challenges by designing and testing new product or service ideas. They will learn and apply the design thinking process alongside core principles from other foundational approaches to innovation and entrepreneurship, such as Design Sprint and the Lean Startup method. Due to the highly interactive, team- and project-based nature of the course, attendance is mandatory (min. 75%)</p> <p>M2: The course introduces students to the fundamentals of programming using Python, with a focus on syntax, control structures, and logical reasoning. It also covers essential topics in computational complexity and classical algorithms for solving problems such as sorting, recursion, and combinatorial optimization.</p> <p>The course is designed not to train professional programmers, but to foster a problem-solving mindset and a structured approach to thinking, particularly valuable for future entrepreneurs and project managers.</p>
<b>Themen der Lehrveranstaltung</b>	<p>Course Topics – M1:</p> <ul style="list-style-type: none"> <li>-Design thinking: Definitions and origins</li> <li>-Design thinking process: Emphathize &amp; discover</li> <li>-Design thinking process: Define - Problem framing &amp; reframing</li> <li>-Design thinking process: Ideate &amp; sketch (incl. Design Sprint method, Value Proposition Canvas, Customer journey map, Service blueprint)</li> <li>-Design thinking process: Testing &amp; Experimentation (incl. Lean Startup method)</li> <li>-Pitching an innovation project</li> <li>-Real-world case studies</li> <li>- Digital and physical prototyping</li> </ul> <p>Course Topics – M2:</p> <ol style="list-style-type: none"> <li>1. Introduction to Programming with Python: Basic syntax and program structure. Variables, data types, and operators.</li> <li>Input/output and simple scripts</li> <li>2. Control Structures and Logical Reasoning: Conditional</li> </ol>

	<p>statements (if, if-else, nested conditions). Loops (for, while) and iteration strategies. Logical operators and Boolean reasoning</p> <p>3. Functions and Problem Decomposition: Defining and calling functions. Parameters, return values, and scope. Modular programming and code reusability</p> <p>4. Data Structures for Problem Solving: Lists, tuples, and strings. Sets and dictionaries: properties and use cases. Iterating over structured data</p> <p>5. Algorithmic Thinking: Problem analysis and step-by-step reasoning. Designing algorithms using flowcharts</p> <p>6. Foundations of Computational Complexity: Time and space complexity: intuitive introduction. Big-O notation and growth rates of functions. Practical comparisons of algorithm efficiency</p> <p>7. Classical Algorithms: Sorting algorithms (e.g., Bubble Sort, Merge Sort). Searching strategies (linear search, binary search). Recursion: principles and applications (e.g., factorial, Fibonacci)</p> <p>8. Combinatorial Optimization and Graph Problems: Introduction to optimization problems (e.g., Knapsack problem). Shortest-path algorithms (e.g., Dijkstra's algorithm)</p>
<b>Stichwörter</b>	<p>M1: Design thinking, innovation, problem framing, ideation, prototyping, experimentation</p> <p>M2: Algorithms, Python programming, time complexity, problem-solving.</p>
<b>Empfohlene Voraussetzungen</b>	<p>M1: None.</p> <p>M2: General logical reasoning skills, Basic knowledge of mathematics (e.g., arithmetic, simple algebra, sets)</p>
<b>Propädeutische Lehrveranstaltungen</b>	
<b>Unterrichtsform</b>	<p>M1: Lectures, laboratory activities, company visits, groupwork, individual reflection.</p> <p>In-person attendance is mandatory (min 75%)</p> <p>M2: The course combines lectures with interactive exercises, coding practice in Python, and problem-solving sessions.</p>
<b>Anwesenheitspflicht</b>	M1 Design Thinking & Prototyping - 75% mandatory presence
<b>Spezifische Bildungsziele und erwartete</b>	Intended Learning Outcomes (ILO)

<b>Lernergebnisse</b>	<p><b>ILO 1. KNOWLEDGE AND UNDERSTANDING</b></p> <p>M1 Design Thinking &amp; Prototyping</p> <p>ILO1a: The student acquires advanced knowledge and understanding of the models and tools of economic-business analysis for starting a new company, with particular focus on identifying new market opportunities, accessing and obtaining economic-financial resources, as well as technological and organizational skills for the development of the company.</p> <p>ILO1b: The student acquires advanced knowledge and understanding of models for new product development and innovation management within enterprises</p> <p>ILO1c: The student acquires advanced knowledge and understanding of business analysis tools and solutions for the development of innovations and organisational knowledge</p> <p>M2 Algorithmic thinking and Coding</p> <p>ILO1d: The student acquires knowledge and understanding of the theories and tools of statistical analysis for making market forecasts</p> <p>ILO1e: The student acquires knowledge of quantitative models for the formulation of forecasts necessary to guide management decisions and to predict the life cycle of a product and a sector</p> <p><b>ILO 2. ABILITY TO APPLY KNOWLEDGE AND UNDERSTANDING</b></p> <p>M1 Design Thinking &amp; Prototyping and M2 Algorithmic thinking and Coding</p> <p>ILO2a: ability to acquire and select information that may be relevant from an entrepreneurial point of view, also in economic-productive contexts different from those studied</p> <p>ILO2b: ability to acquire and select relevant information to frame cases of innovation (product, service, social, managerial organisational), also different from the studied contexts</p> <p>M1 Design Thinking &amp; Prototyping</p> <p>ILO2c: Ability to evaluate the entrepreneurial potential associated with the development of an innovation by a company (learning area 2).</p>
-----------------------	--

	<p>M2 Algorithmic thinking and Coding</p> <p>ILO2d: ability to analyse the combination of market opportunities and resources of the enterprise and to identify entrepreneurial formulas, also with the elaboration of original, compatible and sustainable solutions and combinations</p> <p>ILO2e: ability to select product development models, suitable to appropriately analyse a specific economic-productive context</p> <p><b>ILO 3. AUTONOMY OF JUDGEMENT</b></p> <p>M1 Design Thinking &amp; Prototyping and M2 Algorithmic thinking and Coding</p> <p>ILO3a: Acquire the ability to analyse complex entrepreneurial problems, such as the elaboration and evaluation of an entrepreneurial project (business plan) or the development of a new product.</p> <p>ILO3b: Autonomy of judgement is developed in the training activities carried out for the preparation of the thesis, as well as in the exercises that accompany the lectures and that involve group discussions and the comparison of individual analyses carried out by students in preparation for the lecture.</p> <p>M2 Algorithmic thinking and Coding</p> <p>ILO3c: Acquire the ability to make predictions, such as analysing the future consequences of entrepreneurial, managerial and operational choices.</p> <p><b>ILO 4. COMMUNICATION SKILLS</b></p> <p>ILO4a: Acquire the ability to describe and communicate in an intercultural context, in a clear and precise manner, problematic situations typical of the management of a new enterprise and the development of innovation, such as, for example, the conditions for the validation of a problem or solution, the prospects and risks associated with a business model or an innovation project. The development of communication competences assumes heterogeneous situations such as, for example, the presence of internal stakeholders (e.g. colleagues, managers, owners), or external stakeholders (e.g. potential investors, suppliers and other business partners) and the ability to sustain an adversarial process.</p> <p>ILO4b: The achievement of these objectives is assessed in the</p>
--	---

	<p>course of the training activities already mentioned, as well as in the discussion of the final thesis.</p> <p><b>ILO 5. LEARNING SKILLS</b></p> <p>ILO5a: Acquire the ability to study independently, to prepare summaries.</p> <p>ILO5b: Acquire the ability to identify thematic connections and to establish relationships between different cases and contexts of analysis</p> <p>ILO5c: Acquire the ability to frame a new problem systematically and to generate appropriate taxonomies.</p> <p>ILO5d: Acquire the ability to develop general models from the phenomena studied.</p>
<p><b>Spezifisches Bildungsziel und erwartete Lernergebnisse (zusätzliche Informationen)</b></p>	<p>M1: By the end of the course, students should:</p> <ul style="list-style-type: none"> <li>- Understand the basic principles of the Design thinking approach to innovation</li> <li>- Acquire the ability and tools to empathize with users and beneficiaries</li> <li>- Develop critical thinking to frame and reframe user problems</li> <li>- Design innovative solutions building on user research and insights</li> <li>- Rapidly prototype a solution and test it with users</li> <li>- Conduct an innovation project (i.e., new product, service, or system) from start to end</li> <li>- Work collectively in heterogeneous teams and distribute workload</li> </ul> <p>M2: By the end of the course, students will:</p> <ul style="list-style-type: none"> <li>-Understand the basic principles of programming and algorithmic thinking.</li> <li>-Apply control structures, functions, and data structures in Python to solve simple computational problems.</li> <li>-Analyze the efficiency of algorithms using fundamental concepts of computational complexity.</li> <li>-Implement and evaluate classical algorithms such as sorting, recursion, and combinatorial optimization.</li> <li>-Develop logical reasoning and structured problem-solving skills transferable to entrepreneurial and managerial contexts.</li> </ul>

<b>Art der Prüfung</b>	<p>M1: The evaluation is structured around the following components:</p> <ul style="list-style-type: none"> <li>-Project work (in teams) on an innovation project addressing the project challenge</li> <li>-Individual written reflection (3 short individual essays submitted at the beginning, middle and end of the course)</li> <li>-In-class participation: active engagement in class and in project reviews</li> </ul> <p>M2:</p> <p><b>Exam Structure</b></p> <p>The final M2 assessment is structured into three modules:</p> <p>Module 1 – Computer-based written exam on Python syntax: programming exercises in Python. This module may be completed during the course as part of a mid-term test.</p> <p>Module 2 – Paper-based written exam on computational complexity and recursion: exercises and multiple-choice questions. This module may also be completed during the course as part of a mid-term test.</p> <p>Module 3 – Paper-based written exam on fundamental computer science algorithms. This module must be completed exclusively during the official exam session.</p> <p>During the official exam session, students are therefore required to complete Module 3, together with any of the first two modules not successfully completed during the mid-term tests.</p>
<b>Bewertungskriterien</b>	<p>M1: The evaluation is structured around the following components:</p> <ul style="list-style-type: none"> <li>-Project work (in teams) on an innovation project addressing the project challenge</li> <li>-Individual written reflection (3 short individual essays submitted at the beginning, middle and end of the course)</li> <li>-In-class participation: active engagement in class and in project reviews</li> </ul> <p>M2: Assessment is based on correctness of code and answers,</p>

	<p>clarity of reasoning, and appropriate use of algorithms and complexity analysis. Grade is the weighted average of the exam's modules, based on the number of lessons' hours.</p> <p>The overall exam mark will be determined by the assessment of the two modules (M1+M2), using proportional weights based on the respective module credits, to calculate the final grade for the Growth Mindset course</p>
<b>Pflichtliteratur</b>	<p>M1: The following readings constitute the core upon which the course is built. Specific readings will be provided as bibliography for each class.</p> <ul style="list-style-type: none"> <li>• Bland, D. J., &amp; Osterwalder, A. (2019). <i>Testing business ideas: A field guide for rapid experimentation</i>. John Wiley &amp; Sons.</li> <li>• Brown, T. (2009). <i>Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation</i>. Harper Collins.</li> <li>• Knapp, J., Zeratsky, J., &amp; Kowitz, B. (2016). <i>Sprint: How to solve big problems and test new ideas in just five days</i>. Simon and Schuster.</li> <li>• Osterwalder, A., Pigneur, Y., Bernarda, G., &amp; Smith, A. (2015). <i>Value proposition design: How to create products and services customers want</i>. John Wiley &amp; Sons.</li> <li>• Verganti, R. (2009). <i>Design driven innovation: changing the rules of competition by radically innovating what things mean</i>. Harvard Business Press.</li> </ul> <p>M2:</p> <p>There is no traditional textbook for this course. The learning material is created and developed during the lessons in the form of a course handbook. This handbook is built on a set of lecture notes (Jupyter notebooks), which are progressively completed during class together with the instructor. Students are expected to actively contribute by adding code, explanations, and personal notes, thus creating a customized and evolving resource that supports both in-class learning and individual study.</p>
<b>Weiterführende Literatur</b>	<p>M1: The following readings constitute additional readings upon which the course is built. Specific readings will be provided as bibliography for each class.</p>

	<p>Blank, S. (2020). <i>The four steps to the epiphany: successful strategies for products that win</i>. John Wiley &amp; Sons.</p> <p>Liedtka, J., Chen, E., Foley, N. &amp; Kester, D. (2024). <i>The experimentation field book: a step-by-step project guide</i>. Columbia University Press.</p> <p>Martin, R. L. (2009). <i>The design of business: Why design thinking is the next competitive advantage</i>. Harvard Business Press.</p> <p>Osterwalder, A., &amp; Pigneur, Y. (2010). <i>Business model generation: a handbook for visionaries, game changers, and challengers</i>. John Wiley &amp; Sons.</p> <p>Ries, E. (2011). <i>The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses</i>. Crown Business.</p> <p>Thiel, P. &amp; Masters, B. (2014). <i>Zero to one: notes on start ups, or how to build the future</i>. Random House.</p> <p>Thomke, S. H. (2020). <i>Experimentation works: The surprising power of business experiments</i>. Harvard Business Press.</p> <p>Verganti, R. (2017). <i>Overcrowded. Designing meaningful products in a world awash with ideas</i>. MIT Press.</p> <p>M2: For those interested in further reading or deepening their understanding, the following books and web resources are recommended (but not required)::</p> <ul style="list-style-type: none"> <li>• Horstmann C.S., Necaise R.D. (2019). <i>Python for Everyone</i>. John Wiley &amp; Sons</li> <li>• Wirth, N. (1986). "Algorithms and data structures". Prentice-Hall.</li> <li>• Official Python documentation: <a href="https://www.python.org/doc/">https://www.python.org/doc/</a></li> </ul>
<b>Weitere Informationen</b>	M2: The course makes use of Anaconda and Jupyter Notebook as the main tools for coding practice and exercises.
<b>Ziele für nachhaltige Entwicklung (SDGs)</b>	Menschenwürdige Arbeit und Wirtschaftswachstum, Hochwertige Bildung

<b>Titel des Bestandteils der Lehrveranstaltung</b>	Design Thinking and Prototyping
<b>Code der Lehrveranstaltung</b>	25555A
<b>Wissenschaftlich-disziplinärer Bereich</b>	CEAR-08/D
<b>Sprache</b>	Englisch
<b>Dozenten/Dozentinnen</b>	dr. Silvia Sanasi, Silvia.Sanasi@unibz.it <a href="https://www.unibz.it/en/faculties/economics-management/academic-staff/person/47276">https://www.unibz.it/en/faculties/economics-management/academic-staff/person/47276</a>
<b>Wissensch. Mitarbeiter/Mitarbeiterin</b>	
<b>Semester</b>	Erstes Semester
<b>KP</b>	6
<b>Verantwortliche/r Dozent/in</b>	
<b>Vorlesungsstunden</b>	36
<b>Laboratoriumsstunden</b>	16
<b>Stunden für individuelles Studium</b>	-
<b>Vorgesehene Sprechzeiten</b>	18
<b>Inhaltsangabe</b>	<p>This project-based course introduces students to design thinking and prototyping as key tools for developing a growth mindset in the context of entrepreneurship and innovation management. Working in diverse teams, students tackle real-world challenges by designing and testing new product or service ideas. They will learn and apply the design thinking process alongside core principles from other foundational approaches to innovation and entrepreneurship, such as Design Sprint and the Lean Startup method. Due to the highly interactive, team- and project-based nature of the course, attendance is mandatory (min. 75%).</p>
<b>Themen der Lehrveranstaltung</b>	
<b>Unterrichtsform</b>	<p>Lectures, laboratory activities, company visits, groupwork, individual reflection.</p> <p>In-person attendance is mandatory (min 75%)</p>

<b>Pflichtliteratur</b>	<p>Bland, D. J., &amp; Osterwalder, A. (2019). <i>Testing business ideas: A field guide for rapid experimentation</i>. John Wiley &amp; Sons.</p> <p>Brown, T. (2009). <i>Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation</i>. Harper Collins.</p> <p>Knapp, J., Zeratsky, J., &amp; Kowitz, B. (2016). <i>Sprint: How to solve big problems and test new ideas in just five days</i>. Simon and Schuster.</p> <p>Osterwalder, A., Pigneur, Y., Bernarda, G., &amp; Smith, A. (2015). <i>Value proposition design: How to create products and services customers want</i>. John Wiley &amp; Sons.</p> <p>Verganti, R. (2009). <i>Design driven innovation: changing the rules of competition by radically innovating what things mean</i>. Harvard Business Press.</p>
<b>Weiterführende Literatur</b>	<p>Blank, S. (2020). <i>The four steps to the epiphany: successful strategies for products that win</i>. John Wiley &amp; Sons.</p> <p>Liedtka, J., Chen, E., Foley, N. &amp; Kester, D. (2024). <i>The experimentation field book: a step-by-step project guide</i>. Columbia University Press.</p> <p>Martin, R. L. (2009). <i>The design of business: Why design thinking is the next competitive advantage</i>. Harvard Business Press.</p> <p>Osterwalder, A., &amp; Pigneur, Y. (2010). <i>Business model generation: a handbook for visionaries, game changers, and challengers</i>. John Wiley &amp; Sons.</p> <p>Ries, E. (2011). <i>The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses</i>. Crown Business.</p> <p>Thiel, P. &amp; Masters, B. (2014). <i>Zero to one: notes on start ups, or how to build the future</i>. Random House.</p> <p>Thomke, S. H. (2020). <i>Experimentation works: The surprising power of business experiments</i>. Harvard Business Press.</p> <p>Verganti, R. (2017). <i>Overcrowded. Designing meaningful products in a world awash with ideas</i>. MIT Press.</p>

## Kursmodul

<b>Titel des Bestandteils der Lehrveranstaltung</b>	Algorithmic Thinking and Coding
---	---------------------------------

<b>Code der Lehrveranstaltung</b>	25555B
<b>Wissenschaftlich-disziplinärer Bereich</b>	IINF-05/A
<b>Sprache</b>	Englisch
<b>Dozenten/Dozentinnen</b>	Dott. Cristina Maria Gangai, CristinaMaria.Gangai@unibz.it <a href="https://www.unibz.it/en/faculties/economics-management/academic-staff/person/50466">https://www.unibz.it/en/faculties/economics-management/academic-staff/person/50466</a>
<b>Wissensch. Mitarbeiter/Mitarbeiterin</b>	
<b>Semester</b>	Erstes Semester
<b>KP</b>	8
<b>Verantwortliche/r Dozent/in</b>	
<b>Vorlesungsstunden</b>	48
<b>Laboratoriumsstunden</b>	-
<b>Stunden für individuelles Studium</b>	-
<b>Vorgesehene Sprechzeiten</b>	24
<b>Inhaltsangabe</b>	<p>The course introduces students to the fundamentals of programming using Python, with a focus on syntax, control structures, and logical reasoning. It also covers essential topics in computational complexity and classical algorithms for solving problems such as sorting, recursion, and combinatorial optimization.</p> <p>The course is designed not to train professional programmers, but to foster a problem-solving mindset and a structured approach to thinking, particularly valuable for future entrepreneurs and project managers.</p>
<b>Themen der Lehrveranstaltung</b>	<p>Course Topics – M2:</p> <ol style="list-style-type: none"> <li>1. Introduction to Programming with Python: Basic syntax and program structure. Variables, data types, and operators. Input/output and simple scripts</li> <li>2. Control Structures and Logical Reasoning: Conditional statements (if, if-else, nested conditions). Loops (for, while) and iteration strategies. Logical operators and Boolean reasoning</li> </ol>

	<p>3. Functions and Problem Decomposition: Defining and calling functions. Parameters, return values, and scope. Modular programming and code reusability</p> <p>4. Data Structures for Problem Solving: Lists, tuples, and strings.</p> <p>Sets and dictionaries: properties and use cases. Iterating over structured data</p> <p>5. Algorithmic Thinking: Problem analysis and step-by-step reasoning. Designing algorithms using flowcharts</p> <p>6. Foundations of Computational Complexity: Time and space complexity: intuitive introduction. Big-O notation and growth rates of functions. Practical comparisons of algorithm efficiency</p> <p>7. Classical Algorithms: Sorting algorithms (e.g., Bubble Sort, Merge Sort). Searching strategies (linear search, binary search). Recursion: principles and applications</p> <p>8. Combinatorial Optimization and Graph Problems: Introduction to optimization problems (e.g., Knapsack problem). Shortest-path algorithms (e.g., Dijkstra's algorithm)</p>
<b>Unterrichtsform</b>	M2: The course combines lectures with interactive exercises, coding practice in Python, and problem-solving sessions.
<b>Pflichtliteratur</b>	M2:  There is no traditional textbook for this course. The learning material is created and developed during the lessons in the form of a course handbook. This handbook is built on a set of lecture notes (Jupyter notebooks), which are progressively completed during class together with the instructor. Students are expected to actively contribute by adding code, explanations, and personal notes, thus creating a customized and evolving resource that supports both in-class learning and individual study.
<b>Weiterführende Literatur</b>	M2: For those interested in further reading or deepening their understanding, the following books and web resources are recommended (but not required)::

	<ul style="list-style-type: none"><li>• Horstmann C.S., Necaise R.D. (2019). <i>Python for Everyone</i>. John Wiley &amp; Sons</li><li>• Wirth, N. (1986). "Algorithms and data structures". Prentice-Hall.</li><li>• Official Python documentation: <a href="https://www.python.org/doc/">https://www.python.org/doc/</a></li></ul>
--	--