Syllabus

Course Description

Course Title	Growth Mindset
Course Code	25555
Course Title Additional	
Scientific-Disciplinary Sector	
Language	English
Degree Course	Master in Entrepreneurship and Innovation
Other Degree Courses (Loaned)	
Lecturers	Dott. Cristina Maria Gangai, CristinaMaria.Gangai@unibz.it https://www.unibz.it/en/faculties/economics-management/academic-staff/person/50466 Dr. Björn Schäfer, Bjoern.Schaefer@unibz.it https://www.unibz.it/en/faculties/economics-management/academic-staff/person/45368 Dr. PhD. Ángeles Munoz-Fernández, Angeles.MunozFernandez@unibz.it https://www.unibz.it/en/faculties/economics-management/academic-staff/person/47683
Teaching Assistant	
Semester	First semester
Course Year/s	1
CP	14
Teaching Hours	Module 1 – 36 hours Module 2 – 48 hours
Lab Hours	Module 1 - 16 hours Module 2 - none
Individual Study Hours	-
Planned Office Hours	42
Contents Summary	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%) 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. 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.
Course Topics	Course Topics – M1: -Design thinking: Definitions and origins -Design thinking process: Emphathize & discover -Design thinking process: Define - Problem framing & reframing -Design thinking process: Ideate & sketch (incl. Design Sprint method, Value Proposition Canvas, Customer journey map, Service blueprint) -Design thinking process: Testing & Experimentation (incl. Lean Startup method) -Pitching an innovation project -Real-world case studies - Digital and physical prototyping Course Topics – M2: 1. Introduction to Programming with Python: Basic syntax and program structure. Variables, data types, and operators. Input/output and simple scripts 2. Control Structures and Logical Reasoning: Conditional statements (if, if-else, nested conditions). Loops (for, while) and

	iteration strategies. Logical operators and Boolean reasoning 3. Functions and Problem Decomposition: Defining and calling functions. Parameters, return values, and scope. Modular programming and code reusability 4. Data Structures for Problem Solving: Lists, tuples, and strings. Sets and dictionaries: properties and use cases. Iterating over structured data 5. Algorithmic Thinking: Problem analysis and step-by-step reasoning. Designing algorithms using flowcharts 6. Foundations of Computational Complexity: Time and space complexity: intuitive introduction. Big-O notation and growth rates of functions. Practical comparisons of algorithm efficiency 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) 8. Combinatorial Optimization and Graph Problems: Introduction to optimization problems (e.g., Knapsack problem). Shortest-path algorithms (e.g., Dijkstra's algorithm)
Keywords	M1: Design thinking, innovation, problem framing, ideation, prototyping, experimentation M2: Algorithms, Python programming, time complexity, problem-solving.
Recommended Prerequisites	M1: None. M2: General logical reasoning skills, Basic knowledge of mathematics (e.g., arithmetic, simple algebra, sets)
Propaedeutic Courses	
Teaching Format	M1: Lectures, laboratory activities, company visits, groupwork, individual reflection. In-person attendance is mandatory (min 75%) M2: The course combines lectures with interactive exercises, coding practice in Python, and problem-solving sessions.
Mandatory Attendance	M1 Design Thinking & Prototyping - 75% mandatory presence
Specific Educational Objectives and Learning Outcomes	Intended Learning Outcomes (ILO) ILO 1. KNOWLEDGE AND UNDERSTANDING M1 Design Thinking & Prototyping

	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. ILO1b: The student acquires advanced knowledge and understanding of models for new product development and innovation management within enterprises ILO1c: The student acquires advanced knowledge and understanding of business analysis tools and solutions for the development of innovations and organisational knowledge M2 Algorithmic thinking and Coding ILO1d: The student acquires knowledge and understanding of the theories and tools of statistical analysis for making market forecasts lLO1e: 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 ILO 2. ABILITY TO APPLY KNOWLEDGE AND UNDERSTANDING M1 Design Thinking & Prototyping and M2 Algorithmic thinking and Coding 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 ILO2b: ability to acquire and select relevant information to frame cases of innovation (product, service, social, managerial organisational), also different from the studied contexts M1 Design Thinking & Prototyping ILO2c: Ability to evaluate the entrepreneurial potential associated with the development of an innovation by a company (learning area 2). M2 Algorithmic thinking and Coding 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 ILO2e: ability to select product development models, suitable to appropriately analyse a specific economic-productive context ILO 3. AUTONOMY OF JUDGEMENT M1 Design Thinking & Prototyping and M2 Algorithmic thinking and Coding 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. 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. M2 Algorithmic thinking and Coding ILO3c: Acquire the ability to make predictions, such as analysing the future consequences of entrepreneurial, managerial and operational choices. ILO 4. COMMUNICATION SKILLS 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. ILO4b: The achievement of these objectives is assessed in the course of the training activities already mentioned, as well as in the discussion of the final thesis.

	ILO 5. LEARNING SKILLS ILO5a: Acquire the ability to study independently, to prepare summaries. ILO5b: Acquire the ability to identify thematic connections and to establish relationships between different cases and contexts of analysis ILO5c: Acquire the ability to frame a new problem systematically and to generate appropriate taxonomies. ILO5d: Acquire the ability to develop general models from the phenomena studied.
Specific Educational Objectives and Learning Outcomes (additional info.)	M1: By the end of the course, students should: - Understand the basic principles of the Design thinking approach to innovation - Acquire the ability and tools to empathize with users and beneficiaries - Develop critical thinking to frame and reframe user problems - Design innovative solutions building on user research and insights - Rapidly prototype a solution and test it with users - Conduct an innovation project (i.e., new product, service, or system) from start to end - Work collectively in heterogeneous teams and distribute workload M2: By the end of the course, students will: -Understand the basic principles of programming and algorithmic thinking. -Apply control structures, functions, and data structures in Python to solve simple computational problems. -Analyze the efficiency of algorithms using fundamental concepts of computational complexity. -Implement and evaluate classical algorithms such as sorting, recursion, and combinatorial optimization. -Develop logical reasoning and structured problem-solving skills transferable to entrepreneurial and managerial contexts.
Assessment	M1: M1 is assessed through coursework only; there is no written exam. Attendance is mandatory (min. 75%).

	Formative assessment (not graded): structured feedback at the project checkpoint (S3) and the problem-fit milestone review (S5, green/amber/red rubric), peer speed-testing of prototypes between teams, and lecturer feedback on test cards and learning cards. Summative assessment (attending students): Team project (60%): an innovation project addressing the course challenge. It consists of the final pitch and prototype (30%), the evidence log with interviews, experiments and learning cards (20%), and the milestone reviews in S5 and S9 (10%). Each student's share is adjusted by peer evaluation (±10%). Assesses ILO 1a, 1b, 1c, 2a, 2b, 2c, 3a, 4a and 5c. Individual written reflection (30%): three short essays (approx. 800 words each, 10% each), submitted at the beginning, middle and end of the course. Assesses ILO 1b, 3b, 5a, 5b and 5d. In-class participation (10%): active contribution to class activities, project reviews and peer critique. Assesses ILO 3b, 4a, 4b and 5b. Non-attending students: because the course is team- and project-based, attendance is mandatory and M1 has no non-attending option. M2: The partial M2 exam is structured into three parts: Part 1 – Computer-based written test on Python syntax and programming: programming exercises assessing students' ability to apply computational thinking, develop solutions to problems, and implement them using Python. Assessed learning outcomes: ILO 2a, ILO 3a, ILO 5c. This part may be completed during the course as part of a mid-term test. Part 2 – Paper-based written test on computational complexity and recursion: exercises and multiple-choice questions assessing students' understanding of computational complexity and their

	ability to analyse algorithmic solutions and apply recursive reasoning. Assessed learning outcomes: ILO 3a, ILO 3b, ILO 5d. This part may also be completed during the course as part of a mid-term test. Part 3 – Paper-based written test on fundamental computer science algorithms: exercises assessing students' knowledge of fundamental algorithms and their ability to analyse and apply appropriate algorithmic approaches to solve problems. Assessed learning outcomes: ILO 3a, ILO 5b, ILO 5c. This part must be completed exclusively during the official exam session. During the official exam session, students are therefore required to complete Part 3, together with any of the first two parts not successfully completed during the mid-term tests. The assessment methods and requirements are the same for attending and non-attending students.
Evaluation Criteria	M1: The evaluation is structured around the following components: -Project work (in teams) on an innovation project addressing the project challenge -Individual written reflection (3 short individual essays submitted at the beginning, middle and end of the course) -In-class participation: active engagement in class and in project reviews M2: Assessment is based on correctness of code and answers, clarity of reasoning, and appropriate use of algorithms and complexity analysis. The final grade is calculated as a weighted average of the three parts of the exam, with weights proportional to the instructional hours allocated to each part: 50% for Part 1, 30% for Part 2, and 20% for Part 3.

	The overall exam mark will be determined by the assessment of the two modules (M1 and M2), using proportional weights based on the respective module credits to calculate the final grade for the Growth Mindset course. A passing grade in both modules is required to pass the course; students must therefore achieve at least 18/30 in each module.
Required Readings	M1: The following readings constitute the core upon which the course is built. Specific readings will be provided as bibliography for each class. • Bland, D. J., & Osterwalder, A. (2019).¿*Testing business ideas: A field guide for rapid experimentation*. John Wiley & Sons. • Brown, T. (2009). *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. Harper Collins. • Knapp, J., Zeratsky, J., & Kowitz, B. (2016).¿*Sprint: How to solve big problems and test new ideas in just five days*. Simon and Schuster. • Osterwalder, A., Pigneur, Y., Bernarda, G., & Smith, A. (2015).¿*Value proposition design: How to create products and services customers want*. John Wiley & Sons. • Verganti, R. (2009).¿*Design driven innovation: changing the rules of competition by radically innovating what things mean*. Harvard Business Press. 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.
Supplementary Readings	M1: The following readings constitute additional readings upon which the course is built. Specific readings will be provided as bibliography for each class. Blank, S. (2020).¿*The four steps to the epiphany: successful strategies for products that win*. John Wiley & Sons.

	Liedtka, J., Chen, E., Foley, N. & Kester, D. (2024).¿*The experimentation field book: a step-by-step project guide*. Columbia University Press. Martin, R. L. (2009).¿*The design of business*: *Why design thinking is the next competitive advantage*. Harvard Business Press. Osterwalder, A., & Pigneur, Y. (2010).¿*Business model generation: a handbook for visionaries, game changers, and challengers*. John Wiley & Sons. Ries, E. (2011). *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses.* Crown Business. Thiel, P. & Masters, B. (2014).¿*Zero to one: notes on start ups, or how to build the future*. Random House. Thomke, S. H. (2020).¿*Experimentation works: The surprising power of business experiments*. Harvard Business Press. Verganti, R. (2017). *Overcrowded. Designing meaningful products in a world awash with ideas.*¿MIT Press. M2: For those interested in further reading or deepening their understanding, the following books and web resources are are recommended (but not required):: • Horstmann C.S., Necaise R.D. (2019). *Python for Everyone.* John Wiley & Sons • Wirth, N. (1986). "Algorithms and data structures". Prentice-Hall. • Official Python documentation: https://www.python.org/doc/
Further Information	M1: M2: The course makes use of Anaconda and Jupyter Notebook as the main tools for coding practice and exercises.
Sustainable Development Goals (SDGs)	Decent work and economic growth, Quality education

Course Module

Course Constituent Title	Design Thinking and Prototyping

Course Code	25555A
Scientific-Disciplinary Sector	CEAR-08/D
Language	English
Lecturers	Dr. Björn Schäfer, Bjoern.Schaefer@unibz.it https://www.unibz.it/en/faculties/economics-management/academic-staff/person/45368 Dr. PhD. Ángeles Munoz-Fernández, Angeles.MunozFernandez@unibz.it https://www.unibz.it/en/faculties/economics-management/academic-staff/person/47683
Teaching Assistant	
Semester	First semester
CP	6
Responsible Lecturer	
Teaching Hours	36
Lab Hours	16
Individual Study Hours	-
Planned Office Hours	18
Contents Summary	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%).
Course Topics	
Teaching Format	Lectures, laboratory activities, company visits, groupwork, individual reflection. Attendance and participation in lectures are mandatory (minimum attendance: 75%); no assessment is available for students who do not meet the attendance requirement.

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

Course Module

Course Constituent Title	Algorithmic Thinking and Coding
Course Code	25555B

Scientific-Disciplinary Sector	IINF-05/A
Language	English
Lecturers	Dott. Cristina Maria Gangai, CristinaMaria.Gangai@unibz.it https://www.unibz.it/en/faculties/economics-management/academic-staff/person/50466
Teaching Assistant	
Semester	First semester
CP	8
Responsible Lecturer	
Teaching Hours	48
Lab Hours	-
Individual Study Hours	-
Planned Office Hours	24
Contents Summary	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. 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.
Course Topics	Course Topics – M2: 1. Introduction to Programming with Python: Basic syntax and program structure. Variables, data types, and operators. Input/output and simple scripts 2. Control Structures and Logical Reasoning: Conditional statements (if, if-else, nested conditions). Loops (for, while) and iteration strategies. Logical operators and Boolean reasoning 3. Functions and Problem Decomposition: Defining and calling functions. Parameters, return values, and scope. Modular programming and code reusability

	4. Data Structures for Problem Solving: Lists, tuples, and strings. Sets and dictionaries: properties and use cases. Iterating over structured data 5. Algorithmic Thinking: Problem analysis and step-by-step reasoning. Designing algorithms using flowcharts 6. Foundations of Computational Complexity: Time and space complexity: intuitive introduction. Big-O notation and growth rates of functions. Practical comparisons of algorithm efficiency 7. Classical Algorithms: Sorting algorithms (e.g., Bubble Sort, Merge Sort). Searching strategies (linear search, binary search). Recursion: principles and applications 8. Combinatorial Optimization and Graph Problems: Introduction to optimization problems (e.g., Knapsack problem). Shortest-path algorithms (e.g., Dijkstra's algorithm)
Teaching Format	M2: The course combines lectures with interactive exercises, coding practice in Python, and problem-solving sessions.
Required Readings	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.
Supplementary Readings	M2: For those interested in further reading or deepening their understanding, the following books and web resources are are recommended (but not required): • Horstmann C.S., Necaise R.D. (2019). *Python for Everyone.* John Wiley & Sons • Wirth, N. (1986). "Algorithms and data structures". Prentice-Hall. • Official Python documentation: https://www.python.org/doc/