Informatica Magistrale (2022)

90720 - Usability & User Experience Design

PDF, ADOC.

Learning outcomes

At the end of the course the student is able to design, implement and evaluate software systems with respect to the dimensions of practicality, experience, affection, meaning and value that they may have on the target audience. Characteristics such as ease of use, usefulness and efficiency are fundamental for the positive evaluation of the user experience of the system. The student will be able to focus the functional analysis of the software on the characteristics and needs of the target audience, will be able to drive the development process so as to guarantee a constant connection between the technical and implementation features and the expectation of the audience, and will be able to evaluate whether and according to which metrics a software satisfies these expectations.

Course contents

Teaching will be mostly in English (some explanations will be repeated in Italian if needed and when asked). The exam will be allowed in both Italian and English. The course content is divided in distinct parts:

Background The evolution of the discipline from Human Computer Interaction to User Experience Design. A description of its scope: the human, the computer, and their interaction.

Usability analysis and design A systematic discussion of the techniques and standards for the management of the process of user experience design, with particular attention to the phases of usability analysis (with and without the participation of users) and the user- and goal-oriented usability design methodologies.

Guidelines, patterns and methods for usability design A discussion, with historical aspects, of the framework on which the concrete aspects of usability design is based. we will also give strong attention to the problem of usability for web applications and mobile apps.

72671 - Complements of Programming Languages

PDF, ADOC.

Learning outcomes

Nowadays the software development requires fast and sophisticated code

transformation and analysis tools. For example, Java code is verified by the Virtual Machine before execution to check basic correctness properties about the memory and the locks that are used. Facebook, before releasing its mobile apps, always submits them

to a tool that finds bugs without running the code. The applications of

performance-critical systems and asset-management systems would be impossible to build

and evolve without compilers that derive correct and optimized machine code from high-level source code.

The objective of this course is to discuss modern code transformation and analysis

techniques and illustrate their implementation. For this reason, the course will

refer to a framework, ANTLR, now widely used in academia and industry to build all sorts of languages, tools, and frameworks. For example, Twitter search uses ANTLR for query parsing, with over 2 billion queries a day.

At the end of the course, the student knows the fundamentals of code transformation

and static analysis. He is able to apply the theory by extending a small, yet expressive and powerful language, by means of the ANTLR framework.

Course contents

Introduction to code transformation and analysis. The ANTLR framework and the corresponding Lexical and Syntactical Analysis. Semantic analysis: symbol table, type checking, static analysis of properties about resource and asset consumptions. Intermediate code generation for standard and advanced features of programming languages. Virtual Machines and Interpreters.

87469 - Decision Making with Constraint Programming

PDF, ADOC.

Learning outcomes

This course covers the fundamental methods in artificial intelligence to model and solve combinatorial problems requiring to take (optimal) decisions in the presence of many complex constraints. Such problems appear often in diverse domains of science, business and industry. The inherent difficulty in solving such practically crucial problems has lead to the development of Constraint Programming (CP), an intelligent decision-support technology. A growing number of companies and research institutions worldwide successfully apply CP technology and contribute to its advancement. Skills in this area are therefore in high demand in the job market. The course combines theoretical foundations with practical modelling and solving of realistic problems. At the end of the course, the student has an understanding of the advanced modelling techniques and efficient solution methods, and possesses the necessary skills for modelling in a CP language and solving with a CP solver.

Course contents

Prerequisites: Basic computer science courses such as discrete mathematics, logic, algorithms and data structures, programming. Prior knowledge on artificial intelligence is not necessary.

Course topics

Overview and successful applications of Constraint Programming (CP)

Modeling techniques

Local consistency notions and constraint propagation

Global constraints

Constructive tree search algorithms, search and propagation, branching heuristics, randomization and search restarts

Optimization problems

Constraint-based scheduling

Heuristic search methods

Hybrid CP and heuristic search methods

Advanced topics and hot research topics in CP

81676 - Digital Forensic

PDF, ADOC.

Learning outcomes

At the end of the course the students will know the main topics of digital forensics. Moreover, they have used several basic tools to manage some common scenarios: single device (computer, tablet, smartphone) and several kinds of file, networking (wireless and wired), e-mail and social media. The students will know the importance of the chain of custody and also the main procedures to acquire, conserve and analyze the data. The students will know both the importance of the final report and the conceptual instruments for its appropriate drafting

Course contents

Understanding the Digital Forensics Profession and Investigations

The Investigator’s Office and Laboratory

Data Acquisition

Processing Crime and Incident Scenes

Working with Windows and CLI Systems

Current Digital Forensics Tools

Linux and Macintosh File Systems

Recovering Graphics Files

Digital Forensics Analysis and Validation

Virtual Machine Forensics, Live Acquisitions, and Network Forensics

E-mail and Social Media Investigations

Mobile Device Forensics

Cloud Forensics

Report Writing for High-Tech Investigations

Expert Testimony in Digital Investigations

Ethics for the Expert Witness

37760 - Systems Simulation

PDF, ADOC.

Learning outcomes

At the end of the course students will have acquired methods and tools to design, implement and validate simulation models for the performance analysis and assessment of computer and communication systems and for the analysis of social systems. Students will be able to design, implement and validate simulation models for the analysis and assessment of complex systems.

Course contents

1. classification of systems and models;
2. analytical models: queueing systems and queueing networks;
3. design of simulation experiments;
4. random number generation and random variate generation;
5. discrete events simulazione techniques;
6. design and implementation of simulators and simulation tools;
7. input and output data analysis of simulation models;
8. verification, validation and testing of simulation models;
9. introduction to parallel and distributed simulation;

10. agent based simulation;
11. simulation and machine learning;
12. digital twin;
13. analysis and evaluation of complex systems.
14. Introduction to Virtual Reality.

90749 - Computing Education

PDF, ADOC.

Learning outcomes

The course aims to provide theoretical knowledge, techniques and tools helpful in teaching computer science. At the end of the course, the student is familiar with the main pedagogical and didactical approaches for teaching computer science at different school levels. Students can organise and teach computer science courses, compare and choose different methodologies to generate teaching materials, and evaluate learning.

Course contents

Learning objectives of this course include:

- knowledge of some historical, epistemological and ethical aspects of Computer Science as a scientific discipline and of the motivations underlying the necessity of its teaching;
- understanding of pedagogical aspects and learning theories in the context of computer science teaching;
- knowledge of multiple CS-specific teaching approaches;
- knowledge of the main cognitive difficulties in learning CS (with particular focus on programming), and knowledge of possible strategies to adopt to overcome them;
- ability to formulate and manage learning paths consistent with national standards and curricula related to Computer Science in schools of all levels;
- planning ability to organize laboratories, classroom equipment; ability to integrate students' devices as useful tools for learning.

Topics covered in the course include:

The scientific vision of Computer Science.

Computational Thinking.

Constructivism and constructionism applied to CS teaching.

Top-down and bottom-up approaches in CS teaching.

Teaching of Programming, Algorithms and Data Structures.

Pedagogical implications in the choice of programming languages.

Teaching of technological aspects: computer architecture, operating systems, networks.

CS teaching methods:

- unplugged
- dramatization/visualization
- code reading exercises
- debugging of others’s code
- program execution visualization
- making/tinkering
- repositories as software museums

Difficulties and misconceptions in learning to program.

Creation of learning paths

- in primary schools
- in lower secondary schools
- in scientific lyceums - applied sciences
- in specific courses of technical institutes
- in other upper secondary schools

Assessment methodologies of computer programs.

Importance, planning, and management of computer laboratories.

- Use of BYOD in lessons.

Software licenses: libre (free and open source) and proprietary software: implications in education

Affective and motivational aspects in computer science learning.

77803 - Service - Oriented Software Engineering

PDF, ADOC.

Learning outcomes

At the end of the course, the student knows the design and implementation of complex software systems using an approach based on

the abstractions of service and process. The student is able to design, model and implement modern virtualized software architectures based on enterprise SOA and/or microservices; the student is also able to model and support the enactment of business processes integrating the services that compose them.

Course contents

Service-oriented architectures (SOAs) are used to build large software systems that operate across multiple organizations (as is the case in the enterprise environment) but also as a basic structure to build flexible and extremely scalable applications (as in the case of microservices).

Designing this class of systems and reasoning about their properties requires the use of appropriate abstractions and modeling techniques.

In this course we will see how service and process abstractions can interoperate to describe complex distributed systems and we will see modeling techniques based on these abstractions that help design them.

We will learn how to model processes (using BPMN), services (using UML) and how to model the interaction between processes and services and between different services through choreographies.

We will see how to exploit these techniques to design applications adhering to the service-oriented architectural style both across-enterprises and through microservices (also by adopting specific patterns).

We will also see in practice the protocols and the technologies that can be used for their implementation; in particular for the technologies to support web services we will see SOAP / WSDL and the RESTful style; for microservices we will see the use of containers (eg Docker), microservice orchestrators (eg Kubernetes), service meshes (eg Linkerd) and serialization technologies (eg Protobuf). As far as microservices are concerned, we will try in particular to understand the strengths and weaknesses of this class of applications and we will understand how to build systems that are both scalable and reliable.

To facilitate the prompt realization of examples of some of the technologies related to web services as they are introduced, we will use the Jolie language which allows rapid prototyping of SOA solutions.

Below is a list of the main topics of the course:

Enterprise software systems

Enterprise architecture and modeling

Business Process Management and BPMN

SOA / Choreography Web services (SOAP / WSLD / RESTful)

Microservices properties, design, patterns and implementation

Jolie

66870 - Concurrent Models and Systems

PDF, ADOC.

Learning outcomes

At the end of the course, the student will learn the basic ingredients of concurrency theory, their models and verification systems on such models. (S)He will be able to analyze simple concurrent programs with automatic or semi-automatic tools.

Course contents

- Introduction to concurrency and to the problem of the correctness of reactive systems design.
- Labeled transition systems
- Behavioral equivalences: traces, simulation, bisimulation f(strong and weak), properties.
- The language CCS: syntax and SOS semantics.
- Subclasses of CCS: finite processes, finite-state processes, regular processes, BPP, finite-net processes, finitary CCS.
- Turing completeness of finitary CCS; undecidability of behavioral equivalences of finitary CCS.
- Value-passing CCS.
- Algebraic properties, behavioral congruences and axiomatizations.
- Espressivieness of CCS: encodability of additional operators (internal choice, hiding, sequential composition)
- The problem of muti-way synchronization: Multi-CCS; case study: dining philosophers.
- Petri nets: definition, equivalences, decidable properties, expressiveness.
- Languages for representing Petri nets. Distributed computability.
- Fixpoint theory, least and greates fixpoints, strong bisimilarity as a greatest fixpoint.
- Hennessy-Milner Logic (HML), estention with recursively defined formulae, modal mu-calculus.
- Analysis and verification tools for CCS and HML: Concurrency Workbench (CWB).
- Modeling, analysis and verification of some mutual exclusion algorithms with CWB.

84401 - Context-Aware Systems

PDF, ADOC.

Learning outcomes

At the end of the course, the student is able to design, deploy and evaluate ubiquitous systems and mobile applications able to adapt their behaviors to the context characteristics and to the current location/activity of the user. At the end of the course, the student: -knows the fundamental concepts of context-aware computing, and the main techniques for the localization of users/devices and the human activity recognition; -knows the fundamental models of context-data representation and managing; - knows the main middleware and software architectures in order to deploy adaptive and ubiquitous applications and services

Course contents

The course addresses the design and deployment of ubiquitous and context-aware services and applications, made possible by the pervasive diffusion on the market of devices able to sense the environment and to analyze the sensed data. The course program is structured in two main parts. The first part illustrates the definition of "context" and context-aware systems, focusing on the design and implementation of location-aware, activity-aware, emotion-aware and social-aware systems. Special focus will be given to the spatial data management, by illustrating the main technologies for indoor/outdoor positioning, mapping APIs, geo-data storage and location intelligence.The second part will present the lifecycle of context-aware systems, focusing on methods and technologies for context acquisition, context modeling and context reasoning. Business seminars will be scheduled during the last week of the course. In the following, we provide a brief summary of the course program:

Introduction and definition of context and context-awareness

Use case of context-aware systems

Location-aware systems

Location-based services

Positioning technologies

Mapping APIs

Spatial database

Location intelligence

Activity-aware systems

Emotion-aware systems and affective computing

Context-aware networking and SDN

Context-aware application components

Context acquisition via primary and secondary sensors

Context modeling (context graphs, context languages, web semantic approaches)

Context reasoning via learning-based or inference-based approaches

Re

23762 - Physics of Complex Systems

PDF, ADOC.

Learning outcomes

Basic knowledge of physical and mathematical methods to develop dynamic and statistical model for the study of complex systems. Basic knowledge of graphical methods 2D and 3D used to illustrate the results.

Course contents

Complex system definition. Role of non-linear interactions.Simple theoretical and numerical models for complex systems.Examples from Physics, economy and biology.Data distribution: comparison between exponential and power laws.Agent, neural network and cellular automata models.

Introduction to the study of dynamical systems with applications to complex systems models. Methods for the study of stochastic dynamical systems and their applications to a statistical mechanics approach. Concept of emergent property, critical state and phase transition. Use of simulations for the study of mathematical physical models.

30214 - Logical basis of Computer Science

PDF, ADOC.

Learning outcomes

At the end of this course, students have acquired the logical foundations of areas of theoretical computer science such as ambda-calculus, rewriting systems, computational complexity, database theory

and formal methods. He is able to encode programming constructs in lambda-calculus, to describe simple algorithms in logical form, to

express queries to databases in predicative form and to specify properties of reactive systems as formulas of temporal logic.

Course contents

First module:

1. Propositional and First-Order Logic (recall)

Syntax, Semantics, Soundness and Completeness, Undecidability of First Order Logic

1. Untyped Lambda Calculus

Syntax and operational semantics. The lambda-calculus as a programming language: evaluation strategies and encodings of data types; Turing completeness

1. Meta-theory of untyped lambda calculus

Confluence.

1. Simply typed lambda-calculus and Curry-Howard isomorphism

Curry’s style and Church’s style syntaxes. Isomorphism with propositional minimal logic. Type checking and type inference algorithms.

1. Meta-theory of simply typed lambda-calculus

Weak and strong normalization theorems

1. Curry-Howard isomorphism and extensions to the typing system

Products and coproduts, empty and singleton types. Parametric polymorphisms (minimal introduction to System-F). Minimal introduction to dependent types and Hindley-Milner polymorphisms.

Second module:

1. Logic and Databases

Relational algebra, FO as Query Language, Cilindrical Algebras, Implementation aspects

1. Logic and Computational Complexity

Finite structures and decision problems. NP and PSPACE characterization via first order logics. SAT Solving.

1. Logic and Formal Methods

LTL and CTL: syntax and semantics. Reactive systems and their verifications. Model Checking.

1. Logic and Artifcial Intelligence

Epistemic logic: syntax, semantics, applications

Readi

70090 - Computer Graphics

PDF, ADOC.

Learning outcomes

At the end of the course, students know fundamentals of 3D computer graphics (polygonal modeling and real-time rendering).

In particular, they are able to model and render scenes making use of suitable open source softwares and libraries.

Course contents

Introduction to two- and three-dimensional computer graphics. The graphics pipeline: modelling and rendering. Fundamentals of input and display devices, 3D scanning. Scan conversion of geometric primitives, two- and three-dimensional transformations and clipping, windowing techniques, three-dimensional viewing and perspective. Basic algorithms for CG clipping, scan-conversion, hidden surface removal, ray tracing. Illumination and color models, local and global shading models, and real-time rendering methods. Polygonal meshes, parametric curves and surfaces, splines and NURBS , subdivision curves and surfaces. Surface reconstructions from 3D data set. Virtual/Augmented Reality. Digital Animation techniques. OpenGL, and 3-D modeling tools. Emphasis is on the development of practical skills in using software Blender, graphics libraries and tools. Programming using C/C++ and OpenGL/GLSL for GPU shader.

12569 - Computational Mathematics

PDF, ADOC.

Learning outcomes

At the end of the course, the student knows varied techniques and tools at the basis of the computational solution of problems from scientific calculus. She/He is able to solve applicative problems, from interdisciplinary study or didactic

areas, in an integrated, symbolic and numeric, software environment.

Course contents

Introduction to the Mathematica environment: Kernel, FrontEnd, notebook.

Introduction to programming within Mathematica.

Graphics and visualization tools.

Employing the system capabilites to analize and solve a particular applied problem, of didactical interest to the student, via a package development.

30216 - Probability Models

PDF, ADOC.

Learning outcomes

At the end of the course the student knows elements of some advanced probability theories with applications to computer science, such as Markov chains with discrete and continuous time. She / he is able to analyze some simple stochastic systems related to applications.

Course contents

Introduction to pricing and hedging of financial derivatives in a one-period market: options, arbitrages, Put-Call parity formula, arbitrage and risk-neutral price, incomplete markets.

Elements of martingale theory: Sigma-algebras and filtrations, conditional expectation, discrete-time stochastic processes, martingales, stopping times, Doob decomposition Th., Markov property, discrete Markov chains.

Pricing and hedging in discrete market models: self-financing and admissible strategies, equivalent martingale measure and First Fundamental Theorem of Asset Pricing, arbitrage-free markets and arbitrage price, completeness and Second Fundamental Theorem of Asset Pricing.

Binomial market model: binomial tree, absence of arbitrage and completeness, arbitrage price and hedging strategies, binomial algorithm, stability and convergence to Black-Scholes model, trinomial model and incomplete markets, examples: European and American options.

Elements of stochastic optimal control: introduction to dynamic programming method.

Elements of supervised (machine) learning: input, output and training sets; hypothesis class; expected loss and empirical risk; introduction to neural networks; deterministic and stochastic gradient descent.

Prerequisites: probability theory

90749 - Computing Education

PDF, ADOC.

Learning outcomes

The course aims to provide theoretical knowledge, techniques and tools helpful in teaching computer science. At the end of the course, the student is familiar with the main pedagogical and didactical approaches for teaching computer science at different school levels. Students can organise and teach computer science courses, compare and choose different methodologies to generate teaching materials, and evaluate learning.

Course contents

Learning objectives of this course include:

- knowledge of some historical, epistemological and ethical aspects of Computer Science as a scientific discipline and of the motivations underlying the necessity of its teaching;
- understanding of pedagogical aspects and learning theories in the context of computer science teaching;
- knowledge of multiple CS-specific teaching approaches;
- knowledge of the main cognitive difficulties in learning CS (with particular focus on programming), and knowledge of possible strategies to adopt to overcome them;

- ability to formulate and manage learning paths consistent with national standards and curricula related to Computer Science in schools of all levels;
- planning ability to organize laboratories, classroom equipment; ability to integrate students' devices as useful tools for learning.

Topics covered in the course include:

The scientific vision of Computer Science.

Computational Thinking.

Constructivism and constructionism applied to CS teaching.

Top-down and bottom-up approaches in CS teaching.

Teaching of Programming, Algorithms and Data Structures.

Pedagogical implications in the choice of programming languages.

Teaching of technological aspects: computer architecture, operating systems, networks.

CS teaching methods:

- unplugged
- dramatization/visualization
- code reading exercises
- debugging of others’s code
- program execution visualization
- making/tinkering
- repositories as software museums

Difficulties and misconceptions in learning to program.

Creation of learning paths

- in primary schools
- in lower secondary schools
- in scientific lyceums - applied sciences
- in specific courses of technical institutes
- in other upper secondary schools

Assessment methodologies of computer programs.

Importance, planning, and management of computer laboratories.

- Use of BYOD in lessons.

Software licenses: libre (free and open source) and proprietary software: implications in education

Affective and motivational aspects in computer science learning.

66870 - Concurrent Models and Systems

PDF, ADOC.

Learning outcomes

At the end of the course, the student will learn the basic ingredients of concurrency theory, their models and verification systems on such models. (S)He will be able to analyze simple concurrent programs with automatic or semi-automatic tools.

Course contents

- Introduction to concurrency and to the problem of the correctness of reactive systems design.
- Labeled transition systems
- Behavioral equivalences: traces, simulation, bisimulation f(strong and weak), properties.
- The language CCS: syntax and SOS semantics.
- Subclasses of CCS: finite processes, finite-state processes, regular processes, BPP, finite-net processes, finitary CCS.
- Turing completeness of finitary CCS; undecidability of behavioral equivalences of finitary CCS.
- Value-passing CCS.
- Algebraic properties, behavioral congruences and axiomatizations.
- Espressivieness of CCS: encodability of additional operators (internal choice, hiding, sequential composition)
- The problem of muti-way synchronization: Multi-CCS; case study: dining philosophers.
- Petri nets: definition, equivalences, decidable properties, expressiveness.
- Languages for representing Petri nets. Distributed computability.
- Fixpoint theory, least and greates fixpoints, strong bisimilarity as a greatest fixpoint.
- Hennessy-Milner Logic (HML), estention with recursively defined formulae, modal mu-calculus.
- Analysis and verification tools for CCS and HML: Concurrency Workbench (CWB).
- Modeling, analysis and verification of some mutual exclusion algorithms with CWB.

84401 - Context-Aware Systems

PDF, ADOC.

Learning outcomes

At the end of the course, the student is able to design, deploy and evaluate ubiquitous systems and mobile applications able to adapt their behaviors to the context characteristics and to the current location/activity of the user. At the end of the course, the student: -knows the fundamental concepts of context-aware computing, and the main techniques for the localization of users/devices and the human activity recognition; -knows the fundamental models of context-data representation and managing; - knows the main middleware and software architectures in order to deploy adaptive and ubiquitous applications and services

Course contents

The course addresses the design and deployment of ubiquitous and context-aware services and applications, made possible by the pervasive diffusion on the market of devices able to sense the environment and to analyze the sensed data. The course program is structured in two main parts. The first part illustrates the definition of "context" and context-aware systems, focusing on the design and implementation of location-aware, activity-aware, emotion-aware and social-aware systems. Special focus will be given to the spatial data management, by illustrating the main technologies for indoor/outdoor positioning, mapping APIs, geo-data storage and location intelligence.The second part will present the lifecycle of context-aware systems, focusing on methods and technologies for context acquisition, context modeling and context reasoning. Business seminars will be scheduled during the last week of the course. In the following, we provide a brief summary of the course program:

Introduction and definition of context and context-awareness

Use case of context-aware systems

Location-aware systems

Location-based services

Positioning technologies

Mapping APIs

Spatial database

Location intelligence

Activity-aware systems

Emotion-aware systems and affective computing

Context-aware networking and SDN

Context-aware application components

Context acquisition via primary and secondary sensors

Context modeling (context graphs, context languages, web semantic approaches)

Context reasoning via learning-based or inference-based approaches

Re

73387 - Creativity and Innovation M

PDF, ADOC.

Learning outcomes

At the end of the course the student will gain knowledge of the following topics. Learnings from the history of science. Theoretical foundations of creative thinking. Cognitive modelling. The DIMAI model. Strategies and processes for specific thinking stages. Innovation: hurdles and strategies for success. Application to study cases.

Course contents

1) The necessity for creativity and its definition

2) Creativity in the history of art and science.

3) Theoretical foundations of creative thinking. Cognitive modelling.

4) The Da Vinci thinking model. Strategies and processes for specific thinking stages.

5) Application of creative thinking to study cases.

86716 - Entrepreneurship - Bologna

PDF, ADOC.

Learning outcomes

The main aim of the course is to give a practical orientation to entrepreneurship through the development of the personal capabilities, leadership and the supply of operational and conceptual tools for the launching of an entrepreneurial innovative venture.

Course contents

Entrepreneurship and leadership

Business Model Canvas: how to develop your business idea

Principles of Marketing and Brand Management

Public Speaking and pitch: how to present a project

Project Management: basic principles

Corporate and social: B corp and benefit companies.

Women’s businesses

How to be supported and what are the initiatives they support and the UNIBO ecosystem.

86673 - Soft Skills to be Effective at Work - Bologna 3

PDF, ADOC.

Learning outcomes

The training course on soft skills aims to create awareness of the importance of soft skills in working contexts and the most appropriate strategies for their development and/or consolidation in university students. These skills are actually essential means, in combination with the specific knowledge and skills of the disciplines in which participants have chosen to specialize, to deal as effectively as possible with the challenges of their future profession and, even before, the steps functional to the entry into the world of work (for example, selection procedures).

Specific Objectives:

The objectives of the course already described are declined in the following specific objectives:

Acquire awareness of the value of soft skills for job placement

Recognize soft skills and their applicability to different contexts and circumstances

Develop the ability to self-assess their soft skills

Encourage the acquisition and/or enhancement of soft skills

Acquire the ability to accurately define an action plan aimed at developing the weakest skills

Course contents

Lessons (18 hours) will be focused on the following three areas:

Analysing/understanding the situation, which includes the ability to gather information/data, to analyse and interpret circumstances and relationships, and to process this information/data in order to identify a proper solution;

Addressing/solving the problems, which is the ability to plan, manage and implement a targeted action plan, to make consistent decisions and to effectively manage changes. This area is further developed through the attendance of the MOOC named "Change management";

Collaborating/interacting with others, which includes the ability of self-presentation, working within teams, interacting effectively with others, being aware of inter-individual dynamics and conflict management.

Moreover, students will be involved in e-learning activities (duration: 6 hours) aimed at encouraging active participation, self-awareness and self-assessment of the soft skills and their importance for job placement.

This e-learning activity will include:

Videos of some interviews with different figures (e.g., HR Directors of large companies) of the importance of soft skills to achieve in the laboru market;

Questionnaires for soft skills self-assessment;

Exercise for reflecting on classroom activities, participation in MOOCs and output of online questionnaires;

A development plan to fill for describing the soft skills ownened, discussing personal career goals and action plans for strengthening the soft skills emerged as crucial for job placement and/or accessing advanced training courses.

The topics will be analyzed/presented in terms of inclusive industrialization, multiculturality, innovation and diversity management too, as indicated by the UN Sustainable Development Goals.

86675 - Soft Skills to be Effective at Work - Bologna 4

PDF, ADOC.

Learning outcomes

The course aims to develop students' awareness concerning the importance of soft skills in work contexts and the most suitable strategies for their enhancement and/or consolidation among university students. These skills, in fact, represent essential means, combined with the knowledge and specific skills of the scientific fields chosen by the participants, to tackle efficiently the challenges involved in their future career path, as well as the steps essential when entering the labour market (for instance, during the recruitment procedure).

Specific aims: The aims of this course already described correspond to the following specific purposes:

Becoming aware of the role played by soft skills when entering the labour market;

Recognizing one’s soft skills and their suitability to different contexts and situations;

Developing the ability to assessment one’s soft skills;

Promoting the development and/or enhancement of one’s soft skills;

Developing the ability to define a specific plan aimed at fostering those skills that seem to be weaker on inadequate.

Course contents

Through the implementation of an active teaching method, based on team works requiring the translation of soft skills in order to face concrete situations similar to typical job search activity (for example, tackling job interviews and in-basket exercises.

Specifically, these exercises will be focused on three areas of expertise:

To read/understand each situation, which includes the ability to gather information/data that allow to analyse and interpret circumstances and relationships, to process this information/data in order to identify a proper solution;

To address/solve the situations, including the ability to plan, manage and implement a targeted action plan, the ability to make consistent decisions and to effectively manage changes. This area of expertise is further developed through the MOOC named "Change management";

To collaborate, which includes the ability to present oneself, to work within teams, to interact effectively with others, being aware of inter-individual dynamics and conflict management. The activities in class (integrated by MOOCs) address specific areas of expertise. Moreover, students will attend a e-learning platform (this activities will take above 6 hours) aimed at encouraging involvement, active participation, awareness of the importance of soft skills for work placement and the ability to self-evaluate one’s own skills.

This e-learning activity will be defined in the following specific contents:

Presentation and description of the role of soft skills through the interviews with different figures pertaining to the labour market (e.g., HR Directors of large companies);

Self-assessment of one’s soft skills;

Integration of this self-assessment activity with classroom activities, participation in MOOCs and the outputs of online questionnaires;

Summary of one’s soft skills. This profile will allow photographing the skills already developed, those that should be enhanced, and a detailed description of a plan aimed at strengthening those skills emerged as critical for job placement activities and/or accessing advanced training courses.

The topics will be analyzed/presented in terms of inclusive industrialization, multiculturality, innovation and diversity management too, as indicated by the UN Sustainable Development Goals.

Re

81799 - Project Management and Soft Skills M

PDF, ADOC.

Learning outcomes

At the end of the course the student will be aware of the main concepts and tools of project management. Morever, they will be able to make an efficient end effective presentation. Finally, they will learn the role played by soft skills in professional activities.

Course contents

The syllabus is reported below. More information on the course approach can be found on www.robertoverdone.org.

I.Introduction (approx 3 hours).

II. A Pragmatic Approach to Project Management (approx 9 hours).

- Projects, Roles, WBS
- Gantt and PERT Diagrams
- Cost Assessment
- Submitting a Project Proposal

III. Soft Skills and Aptitudes (approx 12 hours).

- Introduction to Cognitive Processes
- Soft Skills at Time of Industry 4.0
- Success and Talent
- Aptitudes
- Team Work
- Leadership
- Learning Styles

IV. Communication Skills (approx 9 hours).

- Fundamentals of Human Communication
- Persuasion
- Inter-Personal Communication
- Presentation Skills

Additionally, the course will include a Madness session where all students will introduce themselves in one minute, a CV contest, and, possibly, one or two seminars from professionals.