

# Testi del Syllabus

Resp. Did.	<b>BENINI MARCO</b>	<b>Matricola: 031722</b>
Docente	<b>BENINI MARCO, 6 CFU</b>	
Anno offerta:	<b>2022/2023</b>	
Insegnamento:	<b>SCV0314 - LOGICA</b>	
Corso di studio:	<b>F004 - INFORMATICA</b>	
Anno regolamento:	<b>2021</b>	
CFU:	<b>6</b>	
Settore:	<b>MAT/01</b>	
Tipo Attività:	<b>C - Affine/Integrativa</b>	
Anno corso:	<b>2</b>	
Periodo:	<b>Secondo Semestre</b>	
Sede:	<b>Como - Università degli Studi dell'Insubria</b>	



## Testi in italiano

<b>Lingua insegnamento</b>	Italiano
<b>Obiettivi formativi</b>	<p>Il corso si propone di fornire le conoscenze dei meccanismi di base dell'inferenza logica, attraverso lo studio delle fondamentali nozioni di logica proposizionale classica e del primo ordine. Tali conoscenze sono rivolte a formare e ad aumentare la capacità di astrazione delle informazioni attraverso la rappresentazione simbolica e quindi la capacità della comprensione di un linguaggio scientifico astratto. Verranno accennati alcuni approfondimenti su strumenti di carattere più applicativo come la correttezza dei programmi e la programmazione funzionale.</p> <p>I risultati di apprendimento attesi comprendono la capacità di saper individuare eventuali errori in una argomentazione matematica, e di avere una proprietà di linguaggio tale da poter enunciare un teorema e descrivere una sua dimostrazione. Nello specifico, al termine del corso ci si attende lo studente abbia acquisito le seguenti abilità:</p> <ol style="list-style-type: none"><li>1. effettuare semplici dimostrazioni matematiche in un sistema formale tra quelli presentati.</li><li>2. effettuare dimostrazioni sulle proprietà di un sistema formale tra quelli presentati.</li><li>3. relazionare dimostrazioni e computazioni.</li></ol> <p>La conoscenza dei meccanismi logici del ragionamento matematico permette l'acquisizione di adeguate capacità per l'approfondimento delle proprie conoscenze e per lo sviluppo individuale di nuove competenze.</p>
<b>Prerequisiti</b>	Per un proficuo apprendimento di questo insegnamento lo studente deve padroneggiare le nozioni matematiche e le tecniche dimostrative impartite nell'insegnamento fondamentale di Algebra e Geometria del primo anno, che dunque costituisce propedeuticità obbligatoria.
<b>Contenuti</b>	A. Introduzione A1. aspetti burocratici e amministrativi, storia della logica, relazione con l'informatica (2 ore)

B. Logica Proposizionale  
B1. concetto di induzione strutturale, sintassi, deduzione naturale (2 ore)  
B2. esempi di dimostrazione formale (2 ore)  
B3. semantica a tavole di verità, insiemi minimali di connettivi, forme normali disgiuntive e congiuntive, soddisfabilità (4 ore)  
B4. algebre di Boole (2 ore)  
B5. teorema di correttezza, schema della dimostrazione e correttezza di semplici programmi (2 ore)  
B6. completezza (2 ore)

C. Logica al Primo Ordine  
C1. Sintassi, sostituzione, calcolo naturale (2 ore)  
C2. esempi di deduzione, esempi di formalizzazione (2 ore)  
C3. semantica alla Tarski, teorema di correttezza (2 ore)  
C4. Risoluzione, unificazione, programmazione logica mediante esempi (4 ore)

D. Teoria dei Tipi Semplici  
D1.  $\lambda$ -calcolo, sintassi, riduzioni (2 ore)  
D2. esempi: tipi di dati in  $\lambda$ -calcolo, esempi: programmazione funzionale (4 ore)  
D3. teoria dei tipi semplici, sintassi, riduzioni (2 ore)  
D4. logica proposizionale intuizionista, sintassi, potenza espressiva (2 ore)  
D5. isomorfismo Curry-Howard, dimostrazioni come programmi, correttezza per costruzione, normalizzazione forte, terminazione di programmi (2 ore)

E. Risultati limitativi  
E1. teorema di compattezza, esempi di modelli non standard (2 ore)  
E2. aritmetica di Peano, codifica (2 ore)  
E3. lemma di punto fisso, teorema di incompletezza di Gödel (2 ore)  
E4. funzioni calcolabili, tesi di Church-Turing, teorema di Cantor, esistenza di funzioni incalcolabili (2 ore)  
E5. incalcolabilità e incompletezza, esempi: minori di grafi (2 ore)

Rispetto agli obiettivi formativi,

- la parte A1 è specificatamente diretta a formare il necessario linguaggio scientifico per affrontare il corso; tale linguaggio verrà approfondito in tutte le parti successive;
- le parti B1, B3, B5, C1, C3, D4, D5, E2 ed E3 sono principalmente dirette a formare le conoscenze relative all'inferenza logica, con le parti B2 e C2 specificatamente destinate a formare l'abilità di effettuare prove formali;
- le parti B4, B5, B6, C3, D1, D3, D5, E1, E2, E3 ed E5 sono dirette a formare le competenze relative alla capacità di astrazione formale, con qualche sovrapposizione con l'obiettivo precedente ove esso sia funzionale a raccordare le abilità;
- le parti B3, B5, C4, D2, D5 ed E4 sono destinate a consolidare gli obiettivi formativi precedenti in ambito informatico.

## Metodi didattici

Lezione frontale in lingua italiana con l'ausilio di slides e approfondimenti alla lavagna.

Gli obiettivi didattici verranno raggiunti seguendo il seguente schema:

1. ognuno dei cinque capitoli del corso viene introdotto da una discussione generale di inquadramento nelle discipline matematiche e informatiche, con la spiegazione del senso filosofico della materia che si andrà ad affrontare.
2. verranno illustrate le definizioni e i teoremi fondamentali al fine di fornire l'impianto matematico in modo solido e ordinato.
3. esempi di utilizzo accompagneranno i risultati di particolare rilievo, sia per chiarirne il senso, sia per illustrarne l'applicazione.
4. le parti che necessitano (dimostrazioni formali) saranno compendiate da esercizi svolti in classe dal docente, con particolare enfasi sul metodo per la loro risoluzione.
5. al termine di ogni capitolo, si fornisce un sommario dei risultati ottenuti con lo scopo di fornire la prospettiva atta a inquadrarli criticamente sia rispetto al loro senso matematico e informatico, sia in relazione alle conoscenze acquisite in altri corsi.

## **Modalità di verifica dell'apprendimento**

L'obiettivo della prova d'esame è l'accertamento dell'acquisizione delle conoscenze e delle abilità descritte nella sezione "Obiettivi Formativi", valutando il livello di conoscenza e la capacità di applicare le nozioni e le tecniche dimostrative viste a lezione.

L'esame consiste in una prova scritta della durata di due ore articolata secondo il seguente schema: (parte A) due domande aperte relative alla presentazione e applicazione delle nozioni presentate nell'insegnamento (enunciare teoremi o definizione, discussione dei concetti, specializzare un concetto ad un dominio specifico); (parte B) dimostrazione di uno dei teoremi presentati a lezione; (parte C) due esercizi sullo schema di quelli presentati a lezione.

L'attribuzione del voto finale sarà determinata per il 30% dalla conoscenza delle definizioni e dalla capacità di applicarle (parte A), per il 30% dalla conoscenza e comprensione delle dimostrazioni dei teoremi (parte B), per il 40% dalla capacità di applicare le nozioni apprese nel corso alla soluzione degli esercizi (parte C).

Il voto finale è espresso in trentesimi.

## **Altre informazioni**

Il sito web del corso è: <https://marcobenini.me/lectures/logica-informatica/>

Il ricevimento studenti è su appuntamento da concordarsi via email. Il docente risponde solo a email provenienti dall'account universitario del mittente. In caso di necessità, il ricevimento potrà essere effettuato in teleconferenza.

## **Programma esteso**

A. Introduzione

A1. aspetti burocratici e amministrativi, storia della logica, relazione con l'informatica

B. Logica Proposizionale

B1. concetto di induzione strutturale, sintassi, deduzione naturale

B2. esempi di dimostrazione formale

B3. semantica a tavole di verità, insiemi minimali di connettivi, forme normali disgiuntive e congiuntive, soddisfacibilità

B4. algebre di Boole

B5. teorema di correttezza, schema della dimostrazione e correttezza di semplici programmi

B6. Completezza

C. Logica al Primo Ordine

C1. Sintassi, sostituzione, calcolo naturale

C2. esempi di deduzione, esempi di formalizzazione

C3. semantica alla Tarski, teorema di correttezza

C4. Risoluzione, unificazione, programmazione logica mediante esempi

D. Teoria dei Tipi Semplici

D1.  $\lambda$ -calcolo, sintassi, riduzioni

D2. esempi: tipi di dati in  $\lambda$ -calcolo, esempi: programmazione funzionale

D3. teoria dei tipi semplici, sintassi, riduzioni

D4. logica proposizionale intuizionista, sintassi, potenza espressiva

D5. isomorfismo Curry-Howard, dimostrazioni come programmi, correttezza per costruzione, normalizzazione forte, terminazione di programmi

E. Risultati limitativi

E1. teorema di compattezza, esempi di modelli non standard

E2. aritmetica di Peano, codifica

E3. lemma di punto fisso, teorema di incompletezza di Gödel

E4. funzioni calcolabili, tesi di Church-Turing, teorema di Cantor, esistenza di funzioni incalcolabili

E5. incalcolabilità e incompletezza, esempi: minori di grafi

## **Obiettivi per lo sviluppo sostenibile**

### **Codice**

### **Descrizione**



**Testi in inglese**

## Teaching language

Italian

## Learning outcomes

The course aims at providing the knowledge about the basic mechanisms of logical inference, by studying the fundamental notions of classical propositional and first order logic. These pieces of knowledge are directed toward forming and increasing the information abstraction ability using symbolic representations, and thus the ability to understand an abstract scientific language. Some more specialised subjects and tools will be sketched, such as program correctness and functional programming.

The expected learning outcomes are the ability to detect errors and mistakes in a scientific argument, to possess a language competence high enough to state a theorem and describe its proof. Specifically, at the end of the course, the student is expected to acquire the following abilities:

1. to perform simple mathematical proofs in a formal system among the ones explained;
2. to perform proofs about the properties of a formal system among the ones illustrated in the course;
3. to relate proofs and computations.

The knowledge of the logical mechanisms of mathematical reasoning allows to acquire adequate abilities to deepen personal knowledge and to foster the personal development of new competencies.

## Prerequisites

For a proficient learning of this course, the student has to master the mathematical notions and the proving techniques taught in the fundamental course in Algebra and Geometry during the first year, which is, in any case, compulsory to pass before taking this examination.

## Course content

- A. Introduction
  - A1. Administrative and bureaucratic aspects, history of logic, relation with Computer Science. (2 hours)
- B. Propositional Logic
  - B1. Structural induction, syntax, natural deduction (2 hours)
  - B2. Examples of formal derivations
  - B3. Truth table semantics, minimal sets of connectives, disjunctive and conjunctive normal forms, satisfiability (4 hours)
  - B4. Boolean algebras (2 hours)
  - B5. Soundness theorem, schema of the proof, and correctness of simple programs (2 hours)
  - B6. Completeness (2 hours)
- C. First Order Logic
  - C1. Syntax, substitution, natural deduction (2 hours)
  - C2. Examples of derivation, examples of formalisation (2 hours)
  - C3. Tarski semantics, soundness theorem (2 hours)
  - C4. Resolution, unification, logical programming by examples (4 hours)
- D. Simple Theory of Types
  - D1.  $\lambda$ -calculus, reductions (2 hours)
  - D2. Examples: data types in the  $\lambda$ -calculus; functional programming (4 hours)
  - D3. Simple theory of types, syntax, reductions (2 hours)
  - D4. Intuitionistic propositional logic, syntax, expressive power (2 hours)
  - D5. Curry-Howard isomorphism, proofs as programs, correctness by construction, strong normalisation, program termination (2 hours)
- E. Limiting Results
  - E1. Compactness theorem, examples of non-standard models (2 hours)
  - E2. Peano arithmetic, coding (2 hours)
  - E3. Fixed point lemma, Gödel's incompleteness theorem. (2 hours)
  - E4. Computable functions, Church-Turing thesis, Cantor's theorem, existence of non-computable functions (2 hours)
  - E5. Non-computability and incompleteness, example: graph minors (2 hours)

With respect to the learning objectives,

- part A1 is specifically devoted to form the scientific language needed to start the course; this language will be improved in all the other parts;

- parts B1, B3, B5, C1, C3, D4, D5, E2, and E3 are mainly directed to form the knowledge about logical inference, with parts B2 and C2 specifically oriented toward the formation of the ability to perform formal proofs;  
 - parts B4, B5, B6, C3, D1, D3, D5, E1, E2, E3 and E5 are directed to form the competencies about formal abstraction, with some interaction with the previous learning objective when it is needed to link the abilities;  
 - parts B3, B5, C4, D2, D5 and E4 are deputed to strengthen the previous learning objectives in the Computer Science domain.

## Mode of delivery

Conventional frontal lecture in Italian with slides and blackboard.

The teaching objectives are pursued as follows:

1. each of the five chapters of the course will be introduced by a general discussion to relate the topics to the other mathematical and computer science subjects, and to illustrate the philosophical meaning of what is going to be explained.
2. the fundamental definitions and theorems will be illustrated to provide a solid and well organised mathematical framework.
3. examples will complement the particularly relevant results, both to clarify their meaning, and to illustrate their applications.
4. the parts which need (the formal proofs) will be complemented with exercises to be solved in the classroom by the lecturer, with an emphasis on the techniques to solve them.
5. at the end of each chapter, a summary of what has been achieved is given, to provide the right perspective to critically evaluate the results, both with respect to their inner mathematical meaning, and with respect to what has been acquired in other courses.

## Assessment methods and criteria

The objective of the examination is to ascertain the acquiring of the pieces of knowledge and abilities described in the section "Learning Objectives", evaluating the level of knowledge and the ability to apply the notions and the proving techniques seen during the course.

The examination is written, has to be performed in two hours, and it is divided into three parts: (part A) two open questions about the presentation and the application of the notions illustrated in the course (state theorems or definitions, discuss the notions, specialise a concept to a specific domain); (part B) proof of one of the theorems explained during the course; (part C) two exercises similar to the one described during the course.

The final mark will be determined for the 30% from the knowledge of the definitions and the ability to apply them (part A), for the 30% from the knowledge and understanding of the proofs of theorems (part B), for the 40% from the capacity to apply the notions learned in the course to the solution of exercises (part C).

The final mark is expressed in thirtieths.

## More Information

The course website is: <https://marcobenini.me/lectures/logica-informatica/>

Student reception is by appointment to fix by email. The teacher will answer to email from the official university account of the sender only. If needed, the appointment could be done online.

## Detailed course content

A. Introduction

A1. Administrative and bureaucratic aspects, history of logic, relation with Computer Science.

B. Propositional Logic

B1. Structural induction, syntax, natural deduction

B2. Examples of formal derivations

B3. Truth table semantics, minimal sets of connectives, disjunctive and conjunctive normal forms, satisfiability

B4. Boolean algebras

B5. Soundness theorem, schema of the proof, and correctness of simple programs

B6. Completeness

C. First Order Logic

C1. Syntax, substitution, natural deduction

C2. Examples of derivation, examples of formalisation

C3. Tarski semantics, soundness theorem

- C4. Resolution, unification, logical programming by examples
- D. Simple Theory of Types
- D1.  $\lambda$ -calculus, reductions
- D2. Examples: data types in the  $\lambda$ -calculus; functional programming
- D3. Simple theory of types, syntax, reductions
- D4. Intuitionistic propositional logic, syntax, expressive power
- D5. Curry-Howard isomorphism, proofs as programs, correctness by construction, strong normalisation, program termination
- E. Limiting Results
- E1. Compactness theorem, examples of non-standard models
- E2. Peano arithmetic, coding
- E3. Fixed point lemma, Gödel's incompleteness theorem.
- E4. Computable functions, Church-Turing thesis, Cantor's theorem, existence of non-computable functions
- E5. Non-computability and incompleteness, example: graph minors

## Obiettivi per lo sviluppo sostenibile

Codice	Descrizione
--------	-------------