



Principali informazioni sull'insegnamento	
Denominazione dell'insegnamento	Architettura degli Elaboratori e Sistemi Operativi
Corso di studio	Informatica e Comunicazione Digitale
Anno di corso	Primo
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS):	9
SSD	INF/01
Lingua di erogazione	Italiano
Periodo di erogazione	Primo semestre
Obbligo di frequenza	No (la frequenza è fortemente consigliata)

Docente	
Nome e cognome	Ciro Castiello
Indirizzo mail	ciro.castiello@uniba.it
Telefono	080 5442135
Sede	Dipartimento di Informatica
Sede virtuale	
Ricevimento (giorni, orari e modalità)	È possibile concordare incontri di ricevimento fissando un appuntamento (eventualmente via email). Gli incontri possono svolgersi in presenza oppure online.

Syllabus	
Obiettivi formativi	<p>Il corso ha l'obiettivo di far conoscere l'organizzazione e i principi di funzionamento dei calcolatori e dei sistemi operativi.</p> <p>Lo studio dei calcolatori è strutturato per livelli: attraverso processi di astrazione e virtualizzazione, si procede a esaminare strutture sempre più complesse che governano il funzionamento dei sistemi di elaborazione moderni.</p> <p>Il corso offre una panoramica sui principali componenti di base dei sistemi di elaborazione, sulle tecniche di rappresentazione delle informazioni e sull'aritmetica degli elaboratori.</p>
Prerequisiti	Non sono richieste conoscenze preliminari.
Contenuti di insegnamento (Programma)	<p>Livelli di astrazione in un elaboratore</p> <p>Modello dell'elaboratore</p> <p>Evoluzione degli elaboratori</p> <p>Classificazione delle architetture</p> <p>Componenti dell'elaboratore</p> <p>Memorie RAM, cache, ROM</p> <p>Memorie secondarie</p> <p>Periferiche</p> <p>Livello logico-digitale</p> <p>Algebra di Boole e porte logiche</p> <p>Circuiti combinatori</p> <p>Circuiti sequenziali</p> <p>Livello di microarchitettura</p> <p>Implementazione dell'unità di controllo</p> <p>Bus e gestione dell'I/O</p> <p>Livello ISA</p>



	<p>Dati, istruzioni e metodi di indirizzamento Sistema Operativo - Introduzione Sistema Operativo - Gestione dei processi Sistema Operativo - Sincronizzazione dei processi Sistema Operativo - Gestione della memoria Sistema Operativo - Gestione delle periferiche Sistema Operativo - File System</p> <p>Sistemi di numerazione Rappresentazione dei numeri (insiemi N e Z) Rappresentazione dei numeri (insieme R) Codici Realizzazione del computer Hack (dalla porta logica Nand al livello del linguaggio Assembly):</p> <ul style="list-style-type: none"> - Introduzione al linguaggio HDL - Implementazione delle porte logiche elementari - Implementazione dell'ALU - Implementazione dei dispositivi di memoria (registri, RAM, ROM) - Esempi di programmazione in linguaggio macchina Hack - Implementazione del computer Hack - Implementazione dell'Assembler <p>Cenni sul linguaggio Assembly nell'8086</p>
Testi di riferimento	<p>1- Andrew S. Tanenbaum, "Architettura dei calcolatori: Un approccio strutturale", 5° Edizione (o successive), Pearson Education Italia, 2006, ISBN: 9788871922713.</p> <p>2- Noam Nisan, Shimon Schocken, "The Elements of Computing Systems", The MIT Press, 2008, ISBN: 9780262640688.</p> <p>3- Paolo Ancilotti, Maurelio Boari, Anna Ciampolini, Giuseppe Lipari, "Sistemi Operativi", 2° Edizione (o successive), McGraw-Hill, 2008, ISBN: 9788838660696.</p> <p>4- William Stallings, "Architettura ed organizzazione dei calcolatori", 6° Edizione (o successive), Pearson Education Italia, 2004, ISBN: 9788871922010.</p>
Note ai testi di riferimento	<p>1, 2, 3: testi adottati; 4: testo consigliato.</p>

Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Pratica (laboratorio, campo, esercitazione, altro)	Studio individuale
225	56	30	139
CFU/ETCS			
9	7	2	

Metodi didattici	
	<ul style="list-style-type: none"> - Lezioni frontali condotte con l'ausilio di supporti didattici (slide). - Esercitazioni.

Risultati di apprendimento previsti	
--	--



Conoscenza e capacità di comprensione	<ul style="list-style-type: none"> - Acquisizione dei fondamenti della disciplina informatica relativamente all'architettura degli elaboratori e ai sistemi operativi, fissando in particolare gli aspetti essenziali che rimangono inalterati a fronte del cambiamento tecnologico. - Comprensione dell'hardware anche da una prospettiva software, considerando l'uso dei componenti principali dell'elaboratore da parte delle applicazioni software. - Consapevolezza delle possibilità e dei limiti delle tecnologie informatiche. - Comprensione del concetto fondamentale di processo inteso nell'accezione di sequenza di eventi generati dall'esecuzione di un programma.
Conoscenza e capacità di comprensione applicate	<ul style="list-style-type: none"> - Analisi più approfondita della disciplina mediante applicazioni e casi di studio che fanno riferimento, in particolare, alla realizzazione (simulata) di un sistema di elaborazione completo e funzionante, al fine di assimilare e mettere in pratica le competenze teoriche acquisite.
Competenze trasversali	<p>Autonomia di giudizio</p> <ul style="list-style-type: none"> - Gli studenti sono in grado di apprezzare i modelli teorici alla base della realizzazione degli elaboratori e l'approccio classico allo studio dei calcolatori strutturato per livelli. - L'autonomia di giudizio viene acquisita attraverso lo studio e l'interpretazione critica dei testi. - Il raggiungimento dell'adeguata autonomia è verificato attraverso le esercitazioni svolte durante il corso, e con l'esame finale di profitto. <p>Abilità comunicative</p> <ul style="list-style-type: none"> - Gli studenti sono in grado di esporre le tematiche incluse nel programma del corso mediante il lessico specifico della disciplina. <p>Capacità di apprendere in modo autonomo</p> <ul style="list-style-type: none"> - Gli studenti sono in grado di approfondire in autonomia le tematiche incluse nel programma del corso anche ricorrendo a risorse non direttamente coinvolte nella erogazione delle ore di lezione.

Valutazione	
Modalità di verifica dell'apprendimento	<ul style="list-style-type: none"> - Realizzazione (simulata) di un sistema di elaborazione completo e funzionante (mediante l'ausilio del software presentato a lezione e messo a disposizione del docente). - La realizzazione del sistema è propedeutica alla partecipazione all'esame di profitto vero e proprio (che può concretizzarsi in una prova scritta od orale). - Svolgimento di una prova intermedia durante lo svolgimento del corso (con eventuale valenza esonerante in vista degli esami).
Criteri di valutazione	<ul style="list-style-type: none"> - Gli studenti devono realizzare un sistema di elaborazione (in forma simulata) completamente funzionante in ogni singolo elemento. - Gli studenti sono valutati in base al livello di conoscenza riguardante le diverse tematiche incluse nel programma.
Criteri di misurazione dell'apprendimento e di attribuzione del voto finale	<p>La realizzazione pratica è propedeutica alla partecipazione all'esame di profitto (costituisce, quindi, una idoneità).</p> <p>La prova d'esame, se svolta in forma scritta, viene valutata attribuendo uno specifico punteggio ai singoli quesiti.</p> <p>Il voto finale è attribuito in trentesimi.</p> <p>L'esame si intende superato quando il voto è maggiore o uguale a 18.</p>
Altro	<ul style="list-style-type: none"> - Si consiglia la frequenza delle lezioni e lo studio costante durante lo svolgimento del corso (anche al fine di partecipare alla prova intermedia).



	- Si consiglia di non rimandare la partecipazione all'esame. In particolare, distanziare la consegna del software dalla partecipazione all'esame risulta controproducente e annulla l'efficacia didattica legata alla realizzazione simulata del sistema.

ENGLISH VERSION

General information	
Academic subject	Computer Architecture and Operating Systems
Degree course	Informatica e Comunicazione Digitale
Academic Year	First
European Credit Transfer and Accumulation System (ECTS)	9
Language	Italian
Academic calendar (starting and ending date)	First semester
Attendance	No (attendance is strongly recommended)

Professor/ Lecturer	
Name and Surname	Ciro Castiello
E-mail	ciro.castiello@uniba.it
Telephone	080 5442135
Department and address	Department of Informatics
Virtual headquarters	
Tutoring (time and day)	Meetings with students can be organized by appointment (appointments can be arranged by email). Students can be met in person or online.

Syllabus	
Learning Objectives	<p>The course aims to provide an understanding of the organisation and working principles of computers and operating systems.</p> <p>The study of computers is structured in levels: through abstraction and virtualisation, increasingly complex structures governing the operation of modern computing systems are examined.</p> <p>The course provides an overview of the main basic components of computing systems, information representation techniques and computer arithmetic.</p>
Course prerequisites	No prerequisites are needed.
Contents	<ul style="list-style-type: none"> Abstraction levels in computers Computer model Computer evolution Classification of architectures Computer components RAM, cache, ROM Memories Secondary Memories Peripherals Digital logic level Boolean Algebra and logic gates Combinational circuits Sequential circuits Microarchitecture level



	<p>Control unit implementation Bus and I/O ISA level Data, instructions and addressing modes Operating System - Introduction Operating System – Process management Operating System – Process Synchronization Operating System – Memory management Operating System – Peripheral management Operating System – File system</p> <p>Numbering systems Representation of numbers (N and Z sets) Representation of numbers (R set) Codes Realization of the Hack computer (from Nand to Assembly language): - Introduction to the HDL language - Implementation of the basic logic gates - Implementation of the ALU - Implementation of the memory devices (registers, RAM, ROM) - Samples of programming in the Hack language - Implementation of the Hack computer - Implementation of the Assembler</p> <p>Rudiments of the Assembly language in 8086</p>
Books and bibliography	<p>1- Andrew S. Tanenbaum, "Architettura dei calcolatori: Un approccio strutturale", 5° Edition (or following), Pearson Education Italia, 2006, ISBN: 9788871922713.</p> <p>2- Noam Nisan, Shimon Schocken, "The Elements of Computing Systems", The MIT Press, 2008, ISBN: 9780262640688.</p> <p>3- Paolo Ancilotti, Aurelio Boari, Anna Ciampolini, Giuseppe Lipari, "Sistemi Operativi", 2° Edition (or following), McGraw-Hill, 2008, ISBN: 9788838660696.</p> <p>4- William Stallings, "Architettura ed organizzazione dei calcolatori", 6° Edition (or following), Pearson Education Italia, 2004, ISBN: 9788871922010.</p>
Additional materials	

Work schedule			
Total	Lectures	Hands on (Laboratory, working groups, seminars, field trips)	Out-of-class study hours/ Self-study hours
Hours			
225	56	30	139
ECTS			
9	7	2	
Teaching strategy			
<ul style="list-style-type: none"> - Lectures supported by teaching aids (slide). - Exercises. 			
Expected learning outcomes			
Knowledge and understanding on:			
<ul style="list-style-type: none"> - Acquisition of the fundamentals of computer science, with regard to computer architecture and operating systems, focussing mostly on the essential issues that remain unchanged in the face of technological shifts. - Understanding of hardware also from a software perspective, considering the 			



	<p>role of the main computer components while software is running.</p> <ul style="list-style-type: none"> - Awareness of the possibilities and limitations of computer technologies. - Understanding of the basic concept of process as a sequence of events generated by the execution of a computer program.
Applying knowledge and understanding on:	<ul style="list-style-type: none"> - In-depth analysis through applications and case studies referring in particular to the (simulated) realization of a complete working computer system, in order to assimilate and put into practice the theoretical skills acquired.
Soft skills	<p><i>Making informed judgments and choices</i></p> <ul style="list-style-type: none"> - Students are able to appreciate the theoretical models underlying the development of computers and the classical approach to the study of computers structured by levels. - The autonomy of judgment is acquired through the study and critical interpretation of texts. - The achievement of adequate autonomy is verified through the exercises carried out during the course, and through the final exam. <p><i>Communicating knowledge and understanding</i></p> <p>Students are able to explain the topics included in the syllabus using the specific vocabulary of the discipline.</p> <p><i>Capacities to continue learning</i></p> <ul style="list-style-type: none"> - Students are able to deepen autonomously the themes included in the syllabus also by using resources not provided during the lessons.

Assessment and feedback	
Methods of assessment	<ul style="list-style-type: none"> - (Simulated) realization of a complete working computer system (by means of the software introduced during the lessons and made available for students). - The realization of the computer system is preparatory to access the actual exam, which may consist in a written or oral test. - An intermediate test is taken during the course (with possible exemption for examinations).
Evaluation criteria	<ul style="list-style-type: none"> - Students must realize a (simulated) computer system which should prove to be working in each of its components. - Students are assessed on the basis of their level of knowledge regarding the different topics included into the syllabus.
Criteria for assessment and attribution of the final mark	<p>The realization of the computer system represents a qualifying test (idoneità) which must be passed before acceding to the exam.</p> <p>The exam test, when performed in written form, is assessed by assigning a specific mark to the individual questions.</p> <p>The final mark is given in thirtieths.</p> <p>The examination is considered passed when the overall mark is greater than or equal to 18.</p>
Additional information	<ul style="list-style-type: none"> - Students are strongly suggested to attend the lectures and to study constantly during the course (which is also needed to take part in the intermediate test). - Participation in the exam should not be delayed. In particular, separating the delivery of the software from participation in the examination is counterproductive and negates the teaching effectiveness of the simulated implementation of the system.