



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Architettura degli elaboratori e Sistemi Operativi (track cognomi A-L)	
Corso di studio	Informatica e Tecnologie per la Produzione del Software	
Anno Accademico	2024/25	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	9 CFU	
Settore Scientifico Disciplinare	INF/01	
Lingua di erogazione	Italiano	
Anno di corso	Primo	
Periodo di erogazione	1 [^] semestre, le date esatte sono riportate nel manifesto/regolamento	
Obbligo di frequenza	No, ma la frequenza è fortemente raccomandata	
Sito web del corso di studio	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/corsi-di-laurea	

Docente/i	
Nome e cognome	Giovanna Castellano
Indirizzo mail	giovanna.castellano@uniba.it
Telefono	080-5442456
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n. 672, 6 [^] piano.
Sede virtuale	Piattaforma e-learning UNIBA - https://elearning.uniba.it/
Sito web del docente	https://www.uniba.it/it/docenti/castellano-giovanna
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Giovedì 14:00 – 16:00, ma anche previo appuntamento per email

Syllabus	
Obiettivi formativi	Il corso ha l'obiettivo di far conoscere l'organizzazione e i principi di funzionamento degli elaboratori e dei sistemi operativi.



	Gli obiettivi formativi del corso sono: • Far comprendere la struttura e i principi di funzionamento degli elementi fondamentali, sia hardware che software, di un sistema di elaborazione • Far comprendere come la progettazione di tali elementi possa influire sulle prestazioni di un sistema di elaborazione • Far comprendere i principi fondamentali dell'aritmetica dell'elaboratore e le principali tecniche di rappresentazione dell'informazione adottate in un elaboratore. • Far comprendere i principi fondamentali della programmazione di basso livello mediante il linguaggio Assembly. • Far comprendere i principi fondamentali di un sistema operativo
Prerequisiti	Concetti di aritmetica di base. Buona comprensione della lingua inglese.
Contenuti di insegnamento (Programma)	TEORIA (56 ore in totale) 1. Fondamenti 1.1 Visione ad alto livello della struttura di un elaboratore (9 ore) Elaborazione automatica dell'informazione. Rappresentazione digitale. Le strutture di informazione. Struttura a livelli di un sistema di elaborazione. Livelli di macchine e linguaggi. Modello della macchina di Von Neumann. Evoluzione degli elaboratori. 1.2 Componenti di un elaboratore (12 ore) Il processore. Gerarchia di memorie. I Registri. La Memoria centrale. La Memoria cache. Le memorie ROM. Le memorie di massa: dischi magnetici e dischi ottici. Componenti per il trasferimento di informazioni: bus e dispositivi di I/O. Temporizzazione e arbitraggio del bus. 2. Architettura dell'elaboratore 2.1 Livello logico digitale (6 ore) Operatori logici di base. Porte logiche di base. Funzioni logiche e forma canonica disgiuntiva. Circuiti logici di tipo combinatorio. Circuiti logici di tipo sequenziale. 2.2 Livello di microarchitettura (9 ore) Struttura del processore. Unità di calcolo. Unità di controllo e Ciclo di Istruzione. Unità di controllo cablata e unità di controllo microprogrammata. Gestione delle interruzioni. Gestione dei dispositivi di I/O: indirizzamento dei dispositivi di I/O, I/O programmato, I/O guidato da interrupt, DMA. 2.3 Livello ISA (Instruction Set Architecture) (3 ore) Caratteristiche delle istruzioni in linguaggio macchina. Tipi di istruzioni. Formato delle istruzioni. Modalità di indirizzamento. Il livello ISA della famiglia INTEL 80x. 2.4 Livello di linguaggio Assembly (3 ore) Processo di assemblaggio. Processo di linking e loading. 2.5 Architetture avanzate (6 ore) Architetture RISC e CISC. Classificazione di Flynn. Sistemi a singolo processore. Parallelismo a livello di istruzione: il pipelining. Architetture superscalari. Parallelismo a livello di processore: sistemi a multiprocessori, sistemi a multielaboratori. Architetture multi-core. 3. Sistema Operativo 3.1 Struttura di un sistema operativo (2 ore) Il sistema operativo come interfaccia e gestore delle risorse di un sistema di elaborazione. Funzioni di un sistema operativo. Nucleo e chiamate di sistema. Multiprogrammazione. 3.2 Gestione dei processi (3 ore) Definizione di processo. Stati di un processo. Transizione tra gli stati.



	Schedulazione, creazione e distruzione di processi. Algoritmi di schedulazione. 3.3 Gestione della memoria (3 ore) Memoria virtuale. Partizionamento statico e dinamico. Paginazione e segmentazione. ESERCITAZIONI E LABORATORIO (30 ore in totale) Aritmetica dell’elaboratore (12 ore) Sistemi di numerazione posizionali: sistema di numerazione binario, ottale e esadecimale. Rappresentazione binaria dei numeri naturali. Rappresentazione binaria dei numeri relativi. Operazioni in binario con numeri naturali e numeri relativi. Rappresentazione binaria dei numeri reali: la rappresentazione a virgola mobile. Codici binari: codici di caratteri; codici a rilevazione di errore; codici a correzione di errore. Esempi ed esercizi. Introduzione alla programmazione Assembly (16 ore) Linguaggio Assembly del processore Intel 8086: variabili e istruzioni di trasferimento dei dati, istruzioni per la gestione degli Interrupt, istruzioni logico-aritmetiche, istruzioni per il controllo del flusso di esecuzione, procedure e macro. Esempi ed esercizi.			
Testi di riferimento	Testo di riferimento: W. Stallings, Architettura e organizzazione dei calcolatori. Progetto e prestazioni 8^edizione Pearson, 2010 Gli studenti che lo desiderano possono ottenere i testi in prestito dalla Biblioteca. Può convenire verificarne la disponibilità mediante il Sistema Bibliotecario di Ateneo https://opac.uniba.it/easyweb/w8018/index.php? e contattare la biblioteca per concordare il prestito.			
Note ai testi di riferimento	Nel corso delle lezioni il docente illustrerà i concetti con l’ausilio di slide che sintetizzano i contenuti del testo di riferimento. Le slide saranno rese disponibili al termine di ogni lezione sulla piattaforma ADA del dipartimento (v. sopra 'sede virtuale'). Per la parte di teoria, si consiglia di studiare dal testo di riferimento. Per la parte di laboratorio, si consiglia di studiare dal materiale fornito dal docente e di svolgere in autonomia e costantemente tutti gli esercizi svolti durante le esercitazioni in aula. Sulla piattaforma ADA sono disponibili: • slide di supporto utilizzate dal docente durante le lezioni; • esercizi con soluzioni; • alcune tracce di prove scritte di esami precedenti.			
Organizzazione della didattica				
Ore				
Totali	Didattica frontale	Laboratorio/esercitazione	Progetto	Studio individuale
225 ore	56 ore	30 ore	0 ore	139 ore
CFU/ETCS				
9 CFU	7 CFU	2 CFU	0 CFU	



Metodi didattici	
	Il corso sarà organizzato in ore di lezioni frontali condotte con l'ausilio di slide, e ore di esercitazioni guidate, riguardanti il linguaggio binario e il linguaggio Assembly. Tali metodi didattici contribuiscono al raggiungimento dei risultati di apprendimento attesi. Infatti le lezioni frontali sono finalizzate all'acquisizione di conoscenze relative all'elaborazione automatica dell'informazione e al funzionamento dell'elaboratore, mentre le esercitazioni guidate contribuiscono ad acquisire la capacità di applicare tali conoscenze attraverso esercizi sull'uso dei linguaggi di basso livello utilizzati dall'elaboratore (linguaggio binario e linguaggio Assembly). Durante le esercitazioni guidate gli studenti avranno la possibilità di confrontarsi con il docente nonché con i colleghi, sperimentando così forme di studio collaborativo.
Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	• Acquisizione dei fondamenti della disciplina informatica relativamente all'architettura degli elaboratori e ai sistemi operativi, fissando in particolare gli aspetti essenziali dell'elaborazione automatica dell'informazione che rimangono inalterati a fronte del cambiamento tecnologico • Conoscenza delle componenti di un elaboratore sia dal punto di vista funzionale che dal punto di vista strutturale e tecnologico • Conoscenza delle principali funzioni di un sistema operativo. • Conoscenza dell'hardware dell'elaboratore anche da una prospettiva software, considerando l'uso dei componenti principali dell'elaboratore da parte delle applicazioni software. • Consapevolezza delle potenzialità e dei limiti delle moderne architetture degli elaboratori. • Comprensione del concetto fondamentale di processo inteso nell'accezione di sequenza di eventi generate dall'esecuzione di un programma
Conoscenza e capacità di comprensione applicate	Analisi più approfondita della disciplina mediante lo sviluppo di programmi scritti in linguaggio Assembly, al fine di assimilare e mettere in pratica le competenze teoriche acquisite.
Competenze trasversali	Autonomia di giudizio Lo studente dovrà dimostrare di aver acquisito autonomia di giudizio riguardo le potenzialità e le problematiche relative ai moderni sistemi di elaborazione. 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, nonché mediante l'esame finale di profitto. Abilità comunicative Lo studente sarà in grado di illustrare in modo appropriato i principi e le tecniche di base utilizzate nei sistemi di elaborazione e le modalità di esecuzione dei programmi, utilizzando il lessico tecnico, specifico della disciplina.



Capacità di apprendere in modo autonomo

Capacità di apprendere i concetti attraverso una integrazione del materiale fornito dal docente con i contenuti del testo di riferimento.

Valutazione

**Modalità di verifica
dell'apprendimento**

Una prima prova di valutazione intermedia, con valore esonerante, si tiene durante la settimana di interruzione delle lezioni, normalmente collocata intorno alla metà di novembre. La prova consiste in una prova scritta che include domande a risposta multipla chiusa riguardanti gli argomenti del corso trattati sino all'ultima lezione precedente la prova intermedia. La prova contiene anche esercizi per verificare la comprensione degli aspetti più applicativi trattati durante la prima parte del corso, ovvero l'aritmetica degli elaboratori. Il tempo previsto è di norma 1h.30min. Il voto è espresso in trentesimi.

Una seconda prova di valutazione intermedia, con valore esonerante, si tiene immediatamente dopo la fine del corso. La seconda prova può essere sostenuta solo in caso di superamento della prima prova intermedia, e consiste in una prova scritta che include domande a risposta multipla chiusa riguardanti gli argomenti del corso trattati nella seconda parte del corso. La prova contiene anche esercizi per verificare la comprensione degli aspetti più applicativi trattati durante la seconda parte del corso, ovvero la programmazione in linguaggio Assembly. Il tempo previsto è di norma 1h.30min. Il voto è espresso in trentesimi.

Il voto risultante dalla media (arrotondata per eccesso) dei voti ottenuti nelle due prove intermedie viene proposto dal docente per essere verbalizzato in uno degli 8 appelli d'esame previsti per l'A.A in corso. Per partecipare a un esame, gli studenti devono iscriversi all'esame su Esse3.

Lo studente può, se preferisce, integrare l'esame con una prova orale. La votazione finale è in trentesimi. I risultati di valutazione sono pubblicati sul portale Esse3.

Gli studenti che non superano le prove di valutazione intermedie, o che scelgono di non parteciparvi, sostengono l'esame in uno degli appelli previsti durante l'anno accademico. L'esame consiste in una prova scritta che include domande a risposta multipla chiusa riguardanti tutto il programma del corso. La prova scritta contiene anche esercizi per verificare la comprensione degli aspetti più applicativi trattati durante il corso, ovvero l'aritmetica degli elaboratori e la programmazione in linguaggio Assembly. In caso di superamento della prova scritta, lo studente può, se lo desidera, integrare l'esame con una prova orale. Il voto finale conseguito (in trentesimi) viene pubblicato esclusivamente sulla piattaforma Esse3.

Materiali permessi per sostenere la prima prova di valutazione intermedia e la prova scritta d'esame: calcolatrice.

Criteri di valutazione

Conoscenza e capacità di comprensione:

- Capacità di comprendere le domande formulate per la prova scritta e rispondere in maniera pertinente
- Capacità di comprendere le linee guida per lo svolgimento degli esercizi

Conoscenza e capacità di comprensione applicate:

Conoscenza esaustiva degli argomenti oggetto del corso e loro utilizzo nello svolgimento della prova scritta

Autonomia di giudizio:

Capacità di correggere e validare il corretto funzionamento dei programmi scritti in



	linguaggio Assembly. Capacità di apprendere: Comprensione dei contenuti del corso e capacità utilizzare i concetti appresi nello svolgimento di esercizi.																
Criteria di misurazione dell'apprendimento e di attribuzione del voto finale	La prova scritta include 24 domande di teoria a risposta chiusa e/o esercizi su temi inclusi nel programma del corso, e un esercizio di programmazione in linguaggio Assembly. La prova scritta è valutata attribuendo uno specifico punteggio alle singole risposte fornite dallo studente. Ciascuna risposta è valutata 1 punto se corretta, 0 se non data, -0.1 se sbagliata. L'esercizio di programmazione Assembly è valutato sino a un massimo di 6 punti. Il voto finale è attribuito in trentesimi. La prova si ritiene superata se lo studente consegue una votazione di almeno 18/30. <table border="1"> <thead> <tr> <th>Voto</th><th>Descrittori</th></tr> </thead> <tbody> <tr> <td>< 18 insufficiente</td><td>Conoscenze frammentarie e superficiali dei contenuti, errori nell'applicare i concetti, descrizione carente.</td></tr> <tr> <td>18 - 20</td><td>Conoscenze dei contenuti sufficienti ma generali, descrizione semplice, incertezze nell'applicazione di concetti teorici.</td></tr> <tr> <td>21 - 23</td><td>Conoscenze dei contenuti appropriate ma non approfondite, capacità di applicare i concetti teorici, capacità di presentare i contenuti in modo semplice.</td></tr> <tr> <td>24 - 25</td><td>Conoscenze dei contenuti appropriate ed ampie, discreta capacità di applicazione delle conoscenze, capacità di presentare i contenuti in modo articolato.</td></tr> <tr> <td>26 - 27</td><td>Conoscenze dei contenuti precise e complete, buona capacità di applicare le conoscenze, capacità di analisi, descrizione chiara e corretta.</td></tr> <tr> <td>28 - 29</td><td>Conoscenze dei contenuti ampie, complete ed approfondite, buona applicazione dei contenuti, buona capacità di analisi e di sintesi, descrizione sicura e corretta.</td></tr> <tr> <td>30 30 e lode</td><td>Conoscenze dei contenuti molto ampie, complete ed approfondite, capacità ben consolidata di applicare i contenuti, ottima capacità di analisi, di sintesi e di collegamenti interdisciplinari, padronanza di descrizione.</td></tr> </tbody> </table>	Voto	Descrittori	< 18 insufficiente	Conoscenze frammentarie e superficiali dei contenuti, errori nell'applicare i concetti, descrizione carente.	18 - 20	Conoscenze dei contenuti sufficienti ma generali, descrizione semplice, incertezze nell'applicazione di concetti teorici.	21 - 23	Conoscenze dei contenuti appropriate ma non approfondite, capacità di applicare i concetti teorici, capacità di presentare i contenuti in modo semplice.	24 - 25	Conoscenze dei contenuti appropriate ed ampie, discreta capacità di applicazione delle conoscenze, capacità di presentare i contenuti in modo articolato.	26 - 27	Conoscenze dei contenuti precise e complete, buona capacità di applicare le conoscenze, capacità di analisi, descrizione chiara e corretta.	28 - 29	Conoscenze dei contenuti ampie, complete ed approfondite, buona applicazione dei contenuti, buona capacità di analisi e di sintesi, descrizione sicura e corretta.	30 30 e lode	Conoscenze dei contenuti molto ampie, complete ed approfondite, capacità ben consolidata di applicare i contenuti, ottima capacità di analisi, di sintesi e di collegamenti interdisciplinari, padronanza di descrizione.
Voto	Descrittori																
< 18 insufficiente	Conoscenze frammentarie e superficiali dei contenuti, errori nell'applicare i concetti, descrizione carente.																
18 - 20	Conoscenze dei contenuti sufficienti ma generali, descrizione semplice, incertezze nell'applicazione di concetti teorici.																
21 - 23	Conoscenze dei contenuti appropriate ma non approfondite, capacità di applicare i concetti teorici, capacità di presentare i contenuti in modo semplice.																
24 - 25	Conoscenze dei contenuti appropriate ed ampie, discreta capacità di applicazione delle conoscenze, capacità di presentare i contenuti in modo articolato.																
26 - 27	Conoscenze dei contenuti precise e complete, buona capacità di applicare le conoscenze, capacità di analisi, descrizione chiara e corretta.																
28 - 29	Conoscenze dei contenuti ampie, complete ed approfondite, buona applicazione dei contenuti, buona capacità di analisi e di sintesi, descrizione sicura e corretta.																
30 30 e lode	Conoscenze dei contenuti molto ampie, complete ed approfondite, capacità ben consolidata di applicare i contenuti, ottima capacità di analisi, di sintesi e di collegamenti interdisciplinari, padronanza di descrizione.																
Altro	Si suggerisce agli studenti di affidarsi esclusivamente alle informazioni/comunicazioni fornite sui siti ufficiali del Dipartimento di Informatica, ovvero sui gruppi social solo se costituiti e amministrati esclusivamente dai docenti dei relativi insegnamenti: • https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea • https://www.uniba.it/it/ricerca/dipartimenti/informatica • https://elearning.uniba.it/ I programmi di tutti gli insegnamenti sono disponibili al seguente link: • https://elearning.uniba.it/ Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei regolamenti didattici dei Corsi di Studi disponibili nel sito: • https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea Si suggerisce agli studenti di diffidare delle informazioni e dei materiali circolanti																



su siti o gruppi social non ufficiali, poiché spesso sono risultati non affidabili, non corretti o incompleti. Per ogni dubbio, chiedere un incontro al docente secondo le modalità previste per il ricevimento.

Il link al corso sulla piattaforma e-learning UNIBA è il seguente:
<https://elearning.uniba.it/course/view.php?id=2782>



Main information on the course

Course name	Architettura degli elaboratori e Sistemi Operativi (track surnames A-L)	
Degree	Informatica e Tecnologie per la Produzione del Software	
Academic year	2023/24	
European Credit Transfer and Accumulation System (ECTS), in Italian Crediti Formativi Universitari (CFU)	9 CFU (each CFU corresponds to 25 hours (h) of student's time); CFU are of type T1, T2 or T3 T1 = 8 h lecture + 17 h individual study T2 = 15 h practice + 10 h individual study T3 = 25 h individual study	
Settore Scientifico Disciplinare	INF01	
Course language	Italian	
Course year	First	
Course period	First Semester/Second Semester - exact dates can be found in the didactic regulations	
Course attendance requirement	None, but it is highly recommended to attend classes	
Website of the Degree	https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/informatica-tps-270/laurea-triennale-in-informatica-e-tecnologie-per-la-produzione-del-software-d.m.-270	

Teacher(s)

Name and Surname	Giovanna Castellano
email	giovanna.castellano@uniba.it
phone	080-5442456
office	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Room n. 672, 6 th floor.
e-learning platform	ADA - https://elearning.di.uniba.it/
Teacher's homepage	https://www.uniba.it/it/docenti/castellano-giovanna
Office hours	Thursday 14:00 – 16:00, but also upon appointment by email

Syllabus

Course goals	The course is designed to acquaint students with the organization and operating principles of processors and operating systems. The educational objectives of the course are: • To provide an understanding of the structure and operating principles of the fundamental elements, both hardware and software, of a computing system • To provide an understanding of how the design of these elements can affect the performance of a computing system • To provide an understanding of the fundamental principles of computer arithmetic and the main information representation techniques adopted in a computer. • Give an understanding of the basic principles of low-level programming using the Assembly language • Give an understanding of the basic principles of an operating system
Prerequisites/requirements	Basic arithmetic concepts. Good understanding of the English language.



Course program	LECTURES (56 hours total) 1. Fundamentals 1.1 High-level view of the structure of a computer (9 hours) Automatic information processing. Digital representation. Information structures. Layered structure of a computing system. Machine levels and languages. Von Neumann machine model. Evolution of processors. 1.2 Components of a computer (12 hours) The processor. Hierarchy of memories. The Registers. The central memory. The cache memory. The ROM memories. The mass memories: magnetic disks and optical disks. Information transfer components: bus and I/O devices. Bus timing and arbitration. 2. Computer architecture. 2.1 Digital logic level (6 hours) Basic logic operators. Basic logic gates. Logic functions and disjunctive canonical form. Logic circuits of combinatorial type. Logic circuits of sequential type. 2.2 Microarchitecture level (9 hours) Structure of the processor. Computation Unit. Control unit and Instruction Cycle. Wired control unit and microprogrammed control unit. Interrupt management. I/O device management: addressing of I/O devices, programmed I/O, interrupt-driven I/O, DMA. 2.3 Instruction Set Architecture (ISA) level (3 hours) Characteristics of machine language instructions. Types of instructions. Format of instructions. Addressing modes. The ISA level of the INTEL 80x family. 2.4 Assembly language level (3 hours) Assembly process. Linking and loading process. 2.5 Advanced architectures (6 hours) RISC and CISC architectures. Flynn's classification. Single-processor systems. Instruction-level parallelism: pipelining. Superscalar architectures. Processor-level parallelism: multiprocessor systems, multiprocessor systems. Multi-core architectures. 3. Operating System 3.1 Structure of an operating system (2 hours) The operating system as the interface and resource manager of a computing system. Functions of an operating system. Core and system calls. Multiprogramming. 3.2 Process management (3 hours) Definition of a process. States of a process. Transition between states. Scheduling, creation and destruction of processes. Scheduling algorithms. 3.3 Memory management (3 hours) Virtual memory. Static and dynamic partitioning. Paging and segmentation. PRACTICE SESSIONS AND LABORATORY (30 hours total) Computer arithmetic (12 hours) Positional numbering systems: binary, octal and hexadecimal numbering system. Binary representation of natural numbers. Binary representation of relative numbers. Binary operations with natural and relative numbers. Binary representation of real numbers: the floating-point representation. Binary codes: character codes; error-detecting codes; error-correcting codes. Examples and exercises. Introduction to Assembly Programming (16 hours) Assembly language of the Intel 8086 processor: variables and data transfer instructions, Interrupt handling instructions, logic-arithmetic instructions, execution flow control instructions, procedures and macros. Examples and exercises.
Books of reference	Reference book: W. Stallings, Architettura e organizzazione dei calcolatori. Progetto e prestazioni 8^a edizione Pearson, 2010 Students who wish to do so can obtain texts on loan from the library. It may be



	convenient to check their availability through the University Library System https://opac.uniba.it/easyweb/w8018/index.php? and contact the library to arrange borrowing.			
Notes to the books	During the lectures, the teacher will illustrate the concepts with the help of slides summarizing the contents of the reference text. The slides will be made available at the end of each lecture on the department's ADA platform (see 'virtual seat' above). For the theory part, it is recommended to study from the reference book. For the laboratory part, it is recommended to study from the material provided by the teacher and to perform independently and consistently all the exercises done during the practice sessions. Available on the ADA platform are: • supporting slides used by the teacher during lectures; • exercises with solutions; • some traces of written tests from previous exams.			
Organization of the didactic activities				
Hours				
Total	Lectures	Practice sessions	Project work	Individual study
225 hours	56 hours	30 hours	0 hours	139 hours
CFU/ETCS				
9 CFU	7 CFU	2 CFU	0 CFU	

Teaching methods	
	The course will be organized into hours of face-to-face lectures conducted with the aid of slides, and hours of guided exercises, covering binary language and Assembly language. These teaching methods contribute to the achievement of the expected learning outcomes. In fact, the lectures are aimed at acquiring knowledge about automatic information processing and computer operation, while the guided exercises contribute to acquiring the ability to apply this knowledge through exercises on the use of low-level languages used by the computer (binary language and Assembly language). During the guided exercises, students will have the opportunity to engage with the teacher as well as with colleagues, thus experiencing forms of collaborative study.

Expected learning outcomes	
Knowledge and understanding	- Acquisition of the fundamentals of the discipline of computer science with regard to computer architecture and operating systems, fixing in particular the essential aspects of automatic information processing that remain unchanged in the face of technological change - Knowledge of the components of a computer from both functional and structural and technological perspectives - Knowledge of the main functions of an operating system - Knowledge of computer hardware also from a software perspective, considering the use of the main components of the computer by software applications - Awareness of the potential and limitations of modern computer architectures



	- Understanding of the fundamental concept of process understood in the sense of a sequence of events generated by the execution of a program
Applying knowledge and understanding	Deeper analysis of the discipline by developing programs written in Assembly language in order to assimilate and put into practice the theoretical skills acquired.
Other skills	<i>Making judgements</i> The student is expected to demonstrate autonomy of judgment regarding the potential and problems related to modern computing systems. 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, as well as through the final profit examination. <i>Communication</i> The student will be able to appropriately illustrate the basic principles and techniques used in computing systems and how programs are executed, using technical vocabulary, specific to the discipline. <i>Learning skills</i> Ability to learn concepts through an integration of the material provided by the teacher with the contents of the reference text.

Assessment	
Assessment methods	A first middle term partial exam, with exonerating value, is held during the week of class break, normally placed around mid-November. The midterm exam consists of a written test that includes closed multiple choice questions covering the course topics covered up to the last class prior to the midterm. The test also contains exercises to test understanding of the more applied aspects covered during the first part of the course, namely computer arithmetic. The time allowed is normally 1h.30min. The grade is given in thirtieths. A second intermediate assessment test, with exonerating value, is held immediately after the end of the course. The second test can be taken only if the first midterm is passed, and consists of a written test that includes closed multiple-choice questions covering the course topics covered in the second part of the course. The test also contains exercises to test understanding of the more applied aspects covered during the second part of the course, namely programming in Assembly language. The time allowed is normally 1h.30min. The grade is expressed in thirtieths. The grade resulting from the average (rounded up) of the grades obtained in the two midterm exams is proposed by the teacher to be verbalized in one of the 8 exam appointments scheduled for the current A.Y. The student may, if he or she prefers, supplement the exam with an oral test. The final grade is in thirtieths. Grading results are published on the Esse3 portal. Students who do not pass the midterm assessment tests, or who choose not to participate in them, take the exam in one of the official dates of the exam in the academic year. To participate to an exam, students have to make a registration to the exam in ESSE3.



	The exam consists of a written test that includes multiple-choice closed questions covering the entire course syllabus. The written test also contains exercises to test understanding of the more applied aspects covered in the course, namely computer arithmetic and Assembly language programming. If the written test is passed, the student may, if desired, supplement the examination with an oral test. The final grade achieved (in thirtieths) is published exclusively on the Esse3 platform. Materials allowed to take the first midterm and the written examination: calculator.																
Evaluation criteria	Knowledge and Ability to Understand: - Ability to understand the questions formulated for the written test and answer them pertinently - Ability to understand the guidelines for conducting the exercises Applied knowledge and understanding skills: Comprehensive knowledge of the topics covered in the course and their use in the conduct of the written test Autonomy of judgment: Ability to correct and validate the correct operation of programs written in Assembly language. Ability to learn: Understanding of course content and ability to use the concepts learned in the performance of exercises.																
Measurements and final grade	The written test includes 24 closed-ended theory questions and/or exercises on topics included in the course syllabus, and an Assembly language programming exercise. The written test is evaluated by assigning a specific score to the individual answers given by the student. Each answer is scored 1 point if correct, 0 if not given, -0.1 if wrong. The Assembly programming exercise is graded up to a maximum of 6 points. The final grade is given in thirtieths. The test is considered passed if the student achieves a grade of at least 18/30. <table border="1"> <thead> <tr> <th>Grade</th><th>Description</th></tr> </thead> <tbody> <tr> <td>< 18 insufficient</td><td>Fragmentary and superficial knowledge of content, errors in applying concepts, poor description.</td></tr> <tr> <td>18 - 20</td><td>Sufficient but general content knowledge, simple description, uncertainties in applying theoretical concepts.</td></tr> <tr> <td>21 - 23</td><td>Content knowledge appropriate but not in-depth, ability to apply theoretical concepts, ability to present content in a simple way.</td></tr> <tr> <td>24 - 25</td><td>Appropriate and broad content knowledge, fair ability to apply knowledge, ability to present content articulately.</td></tr> <tr> <td>26 - 27</td><td>Accurate and complete content knowledge, good ability to apply knowledge, ability to analyze, clear and correct description.</td></tr> <tr> <td>28 - 29</td><td>Broad, complete and thorough content knowledge, good application of content, good analytical and summarizing skills, confident and correct description.</td></tr> <tr> <td>30 30 with honors</td><td>Very broad, complete and thorough content knowledge, well-established ability to apply content, excellent ability to analyze, synthesize and make interdisciplinary connections, mastery of description.</td></tr> </tbody> </table>	Grade	Description	< 18 insufficient	Fragmentary and superficial knowledge of content, errors in applying concepts, poor description.	18 - 20	Sufficient but general content knowledge, simple description, uncertainties in applying theoretical concepts.	21 - 23	Content knowledge appropriate but not in-depth, ability to apply theoretical concepts, ability to present content in a simple way.	24 - 25	Appropriate and broad content knowledge, fair ability to apply knowledge, ability to present content articulately.	26 - 27	Accurate and complete content knowledge, good ability to apply knowledge, ability to analyze, clear and correct description.	28 - 29	Broad, complete and thorough content knowledge, good application of content, good analytical and summarizing skills, confident and correct description.	30 30 with honors	Very broad, complete and thorough content knowledge, well-established ability to apply content, excellent ability to analyze, synthesize and make interdisciplinary connections, mastery of description.
Grade	Description																
< 18 insufficient	Fragmentary and superficial knowledge of content, errors in applying concepts, poor description.																
18 - 20	Sufficient but general content knowledge, simple description, uncertainties in applying theoretical concepts.																
21 - 23	Content knowledge appropriate but not in-depth, ability to apply theoretical concepts, ability to present content in a simple way.																
24 - 25	Appropriate and broad content knowledge, fair ability to apply knowledge, ability to present content articulately.																
26 - 27	Accurate and complete content knowledge, good ability to apply knowledge, ability to analyze, clear and correct description.																
28 - 29	Broad, complete and thorough content knowledge, good application of content, good analytical and summarizing skills, confident and correct description.																
30 30 with honors	Very broad, complete and thorough content knowledge, well-established ability to apply content, excellent ability to analyze, synthesize and make interdisciplinary connections, mastery of description.																



Further information

It is suggested that students rely exclusively on the information/communication provided on the official websites of the Department of Computer Science, or on social groups only if they are formed and administered exclusively by the faculty members of the relevant subjects:

- <https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea>
- <https://www.uniba.it/it/ricerca/dipartimenti/informatica>
- <https://elearning.uniba.it/>

The syllabi for all the teachings are available at the following link:
<https://elearning.uniba.it/>

The information that all students should know is written in the teaching regulations of the Study Courses available on the website:

- <https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea>

Students are suggested to be wary of information and materials circulating on unofficial sites or social groups, as they are often found to be unreliable, incorrect or incomplete. If you have any doubts, ask for a meeting with the lecturer in accordance with the reception arrangements.

The link to the course on the UNIBA e-learning platform is as follows:
<https://elearning.uniba.it/course/view.php?id=2782>