



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Architettura degli elaboratori e Sistemi Operativi	
Corso di studio	Informatica per la Comunicazione Digitale	
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	
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	

Docenti

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

Docenti

Nome e cognome	Gabriella Casalino
Indirizzo mail	Gabriella.casalino@uniba.it
Telefono	+39 0805712410



Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.674, 6 [^] piano.
Sede virtuale	Piattaforma e-learning UNIBA - https://elearning.uniba.it/
Sito web del docente	https://sites.google.com/site/cilabuniba/people/gabriella-casalino
Ricevimento (giorni, orari e modalità, es. su appuntamento)	Giovedì 14:00 – 16:00, ma anche previo appuntamento per email

Syllabus	
Obiettivi formativi	<p>Il corso ha l'obiettivo di far conoscere l'organizzazione e i principi di funzionamento degli elaboratori e dei sistemi operativi.</p> <p>Gli obiettivi formativi del corso sono:</p> <ul style="list-style-type: none">• 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. <p>Far comprendere i principi fondamentali di un sistema operativo</p>
Prerequisiti	Concetti di aritmetica di base
Contenuti di insegnamento (Programma)	<p>TEORIA (56 h)</p> <p>1. Fondamenti</p> <p>1.1 Visione ad alto livello della struttura di un elaboratore (12 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.</p> <p>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.</p> <p>2.1 Livello ISA (Instruction Set Architecture) (12 ore) Caratteristiche delle istruzioni in linguaggio macchina. Tipi di istruzioni. Formato delle istruzioni. Modalità di indirizzamento. Il livello ISA della famiglia INTEL 80x.</p> <p>2.2 Livello di linguaggio Assembly (2 ore) Processo di assemblaggio. Processo di linking e loading.</p> <p>3. Sistema Operativo</p> <p>3.1 Struttura di un sistema operativo (6 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.</p> <p>3.2 Gestione dei processi (6 ore) Definizione di processo. Stati di un processo. Transizione tra gli stati. Schedulazione, creazione e distruzione di processi. Algoritmi di schedulazione.</p> <p>3.3 Gestione della memoria (6 ore) Memoria virtuale. Partizionamento statico e dinamico. Paginazione e segmentazione.</p> <p>ESERCITAZIONI E LABORATORIO (30 h)</p>



	<p>Aritmetica dell'elaboratore (10 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.</p> <p>Introduzione alla programmazione Assembly (20 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.</p>			
Testi di riferimento	<p><i>G. Bucci, Calcolatori Elettronici. Architettura e organizzazione, Mc Graw Hill, 2017 ISBN: 978-88-3867-546-1</i></p> <p>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.</p>			
Note ai testi di riferimento	<p>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 e-learning. 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.</p> <p>Sulla piattaforma e-learning sono disponibili:</p> <ul style="list-style-type: none"> • slide di supporto utilizzate dal docente durante le lezioni; • esercizi con soluzioni. 			
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	
	<p>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</p>



	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	<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 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	<p>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.</p> <p>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.</p> <p>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</p>

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



	<p>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.</p> <p>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.</p> <p>Il voto risultante dalla media (arrotondata per eccesso) dei voti ottenuti nelle due prove intermedie viene proposto dal docente per essere verbalizzato esclusivamente nella prima sessione di esami (Gennaio/Febbraio). La votazione finale è in trentesimi.</p> <p>Gli studenti che non superano le prove di valutazione intermedie, o che scelgono di non parteciparvi, sostengono l'esame in uno degli appelli previsti a partire da Gennaio. 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. Il voto finale conseguito (in trentesimi) viene pubblicato esclusivamente sulla piattaforma Esse3.</p> <p>Materiali permessi per sostenere la prima prova di valutazione intermedia e la prova scritta d'esame: calcolatrice.</p>								
<p>Criteria di valutazione</p>	<p>Conoscenza e capacità di comprensione:</p> <ul style="list-style-type: none"> - 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 <p>Conoscenza e capacità di comprensione applicate:</p> <p>Conoscenza esaustiva degli argomenti oggetto del corso e loro utilizzo nello svolgimento della prova scritta</p> <p>Autonomia di giudizio:</p> <p>Capacità di correggere e validare il corretto funzionamento dei programmi scritti in linguaggio Assembly.</p> <p>Capacità di apprendere:</p> <p>Comprensione dei contenuti del corso e capacità utilizzare i concetti appresi nello svolgimento di esercizi.</p>								
<p>Criteria di misurazione dell'apprendimento e di attribuzione del voto finale</p>	<p>La prova scritta include 16 domande a risposta chiusa di teoria e/o esercizi su temi inclusi nel programma del corso. La prova scritta è valutata attribuendo uno specifico punteggio alle singole risposte fornite dallo studente. Ciascuna risposta è valutata 2 punti se corretta, 0 se non data, 0 se sbagliata. Il voto finale è attribuito in trentesimi. La prova si ritiene superata se lo studente consegue una votazione di almeno 18/30.</p> <table border="1" data-bbox="528 1832 1441 2145"> <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> </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.
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.								



	<table border="1"><tbody><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>	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.
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	<p>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:</p> <ul style="list-style-type: none">● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea● https://www.uniba.it/it/ricerca/dipartimenti/informatica● https://elearning.uniba.it/ <p>I programmi di tutti gli insegnamenti sono disponibili al seguente link:</p> <ul style="list-style-type: none">● https://elearning.uniba.it/ <p>Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei regolamenti didattici dei Corsi di Studi disponibili nel sito:</p> <ul style="list-style-type: none">● https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea <p>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.</p> <p>Gli studenti potranno iscriversi alla piattaforma e-learning del dipartimento https://elearning.uniba.it/ dove troveranno il materiale del corso.</p>								



Main information on the course

Course name	Computer Architecture and Operating Systems	
Degree	Computer Science and Digital Communication	
Academic year	2024/25	
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	INF/01	
Course language	Italian	
Course year	First	
Course period	First 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/corsi-di-laurea	

Teacher(s)

Name and Surname	Giovanna Castellano
email	giovanna.castellano@uniba.it
phone	080-5442456
office	Department of Computer Science, Via Orabona 4, 70125, Bari. Room n. 672, 6 [^] floor.
e-learning platform	https://elearning.uniba.it/
Teacher's homepage	https://www.uniba.it/it/docenti/castellano-giovanna
Office hours	Thursday 14:00 – 16:00, but also by appointment via email

Teacher(s)

Name and Surname	Gabriella Casalino
email	Gabriella.casalino@uniba.it
phone	+39 0805712410
office	Department of Computer Science, Via Orabona 4, 70125, Bari. Room n.674, 6 [^] floor.
e-learning platform	https://elearning.uniba.it/
Teacher's homepage	https://sites.google.com/site/cilabuniba/people/gabriella-casalino
Office hours	Thursday 14:00 – 16:00, but also by appointment via email

Syllabus

Course goals	<p>The course aims to provide knowledge of the organization and principles of operation of computers and operating systems. The educational objectives of the course are as follows:</p> <ul style="list-style-type: none">• Understand the structure and operating principles of the fundamental components, both hardware and software, of a computing system.• Comprehend how the design of these components can affect the performance of a computing system.
--------------	--



	<ul style="list-style-type: none"> • Grasp the fundamental principles of computer arithmetic and the main techniques for representing information used in a computer. • Understand the basic principles of low-level programming using Assembly language. • Understand the fundamental principles of an operating system.
Prerequisites/requirements	Basic arithmetic concepts
Course Program	<p>Theory (56 h)</p> <p>1.Fundamentals</p> <p>1.1 High-level view of a computer's structure (12 hours) Automatic information processing. Digital representation. Information structures. Layered structure of a computing system. Machine and language levels. Von Neumann machine model. Evolution of computers.</p> <p>1.2 Components of a computer (12 hours) The processor. Memory hierarchy. Registers. Main memory. Cache memory. ROM memory. Mass storage: magnetic disks and optical disks. Components for information transfer: buses and I/O devices. Timing and bus arbitration.</p> <p>2.1 ISA Level (Instruction Set Architecture) (12 hours) Characteristics of machine language instructions. Types of instructions. Instruction format. Addressing modes. The ISA level of the INTEL 80x family.</p> <p>2.2 Assembly Language Level (2 hours) Assembling process. Linking and loading process.</p> <p>3. Operating System</p> <p>3.1 Structure of an operating system (6 hours) The operating system as an interface and resource manager for a computing system. Functions of an operating system. Kernel and system calls. Multiprogramming.</p> <p>3.2 Process Management (6 hours) Definition of a process. States of a process. Transitions between states. Scheduling, creation, and destruction of processes. Scheduling algorithms.</p> <p>3.3 Memory Management (6 hours) Virtual memory. Static and dynamic partitioning. Paging and segmentation.</p> <p>Practice (30 h)</p> <p>Computer Arithmetic (10 hours) Positional number systems: binary, octal, and hexadecimal numbering systems. Binary representation of natural numbers. Binary representation of signed numbers. Binary operations with natural and signed numbers. Binary representation of real numbers: floating-point representation. Binary codes: character codes; error-detecting codes; error-correcting codes. Examples and exercises.</p> <p>Introduction to Assembly Programming (20 hours) Assembly language of the Intel 8086 processor: data transfer variables and instructions, interrupt management instructions, logical-arithmetic instructions, flow control instructions, procedures, and macros. Examples and exercises.</p>
Books of reference	<i>G. Bucci, Calcolatori Elettronici. Architettura e organizzazione, Mc Graw Hill, 2017 ISBN: 978-88-3867-546-1</i>



	<p>Students can borrow texts from the Library by checking availability through the University Library System at https://opac.uniba.it/easyweb/w8018/index.php and contacting the library to arrange the loan.</p>
Notes to the books	<p>During the lectures, the instructor will explain the concepts with the aid of slides that summarize the content of the reference textbook. The slides will be made available at the end of each lecture on the e-learning platform. For the theoretical part, it is recommended to study from the reference textbook. For the laboratory part, it is recommended to study from the material provided by the instructor and to independently and continuously complete all exercises performed during the class sessions.</p> <p>On the e-learning platform, the following are available:</p> <ul style="list-style-type: none">• Supporting slides used by the instructor during the lectures;• Exercises with solutions



Organization of the didactic activities				
Hours				
Total	Lectures	Practice sessions	Project work	Individual study
225	56	30	0	139
CFU/ETCS				
9 CFU	7 CFU	2 CFU	0 CFU	

Teaching methods
<p>The course will be organized into hours of frontal lessons conducted with the aid of slides, and hours of guided exercises related to binary language and Assembly language. These teaching methods contribute to achieving the expected learning outcomes. Indeed, the frontal lessons are aimed at acquiring knowledge related to the automatic processing of information and the functioning of the computer, while the guided exercises help students acquire the ability to apply this knowledge through exercises using the low-level languages used by the computer (binary language and Assembly language). During the guided exercises, students will have the opportunity to interact with the instructor as well as with their peers, thus experiencing forms of collaborative study.</p>

Expected learning outcomes	
<p>Knowledge and understanding</p> <ul style="list-style-type: none"> • Acquisition of the fundamentals of computer science related to computer architecture and operating systems, particularly establishing the essential aspects of automatic information processing that remain unchanged despite technological advancements. • Knowledge of the components of a computer from both a functional and a structural/technological perspective. • Understanding 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 the fundamental concept of a process, seen as a sequence of events generated by the execution of a program. 	
<p>Applying knowledge and understanding</p>	<p>More in-depth analysis of the discipline through the development of programs written in Assembly language, in order to assimilate and put into practice the theoretical skills acquired.</p>
<p>Other skills</p>	<p><i>Making judgements</i> The student must demonstrate having acquired judgment autonomy regarding the potential and challenges of modern processing systems. This autonomy is developed through the study and critical interpretation of texts. The achievement of adequate autonomy is assessed through exercises conducted during the course, as well as through the final examination.</p>



	<p><i>Communication</i> The student will be able to appropriately explain the basic principles and techniques used in processing systems and the methods of program execution, using the technical terminology specific to the discipline.</p> <p><i>Learning skills</i> The ability to learn concepts by integrating the material provided by the instructor with the content of the reference textbook.</p>
--	---

Assessment	
Assessment methods	<p>There are two midterm evaluations for this course. The first midterm takes place during the lecture break week in mid-November and covers topics discussed in the course up to the last lecture before the midterm. It includes a written test with multiple-choice questions and practical exercises on computer arithmetic. The time allocated for this test is 1 hour and 30 minutes, and the grade is out of thirty.</p> <p>The second midterm is held right after the course ends and can only be taken if the first midterm is passed. It assesses the topics discussed in the second part of the course and includes a written test with multiple-choice questions and practical exercises on Assembly language programming. The time allocated for this test is also 1 hour and 30 minutes, and the grade is out of thirty.</p> <p>The final grade is the average of the two midterm grades, rounded up, and it will be proposed by the instructor for the first exam session (January/February). The final grade is out of thirty.</p> <p>Students who do not pass the midterms or prefer not to take them will have to take the exam in January. The exam covers the entire course syllabus and includes multiple-choice questions and practical exercises on computer arithmetic and Assembly language programming. The final grade for this exam will also be out of thirty.</p> <p>For the first midterm and the written exam, only a calculator is permitted.</p>
Evaluation criteria	<p><i>Knowledge and Understanding:</i></p> <ul style="list-style-type: none">• Ability to understand the questions formulated for the written test and respond appropriately.• Ability to understand the guidelines for completing the exercises. <p><i>Applied Knowledge and Understanding:</i></p> <ul style="list-style-type: none">• Comprehensive knowledge of the course topics and their application in the written test. <p><i>Judgment Autonomy:</i></p> <ul style="list-style-type: none">• Ability to correct and validate the proper functioning of programs written in Assembly language.



	<p><i>Learning Ability:</i></p> <ul style="list-style-type: none">• Understanding of the course content and the ability to use the learned concepts in completing exercises.
Measurements and final grade	<p>The midterm evaluation is held during the lecture break week, typically around mid-November. It consists of a written test with multiple-choice questions covering topics discussed in the course up until the last lecture before the midterm, as well as exercises to assess practical understanding. The exam comprises 16 multiple-choice questions on theory and/or exercises related to the course syllabus. Each correct answer is worth 2 points, while no answer or an incorrect answer receives 0 points. The final grade is given out of thirty, and a score of at least 18/30 is required to pass the exam.</p> <p>Here are the grade descriptors:</p> <ul style="list-style-type: none">• < 18: Insufficient - Fragmentary and superficial knowledge of the content, errors in applying concepts, inadequate description.• 18 - 20: Sufficient - General but adequate knowledge of the content, simple description, some uncertainties in applying theoretical concepts.• 21 - 23: Good - Appropriate but not in-depth knowledge of the content, ability to apply theoretical concepts, ability to present content simply.• 24-25: Very Good - Appropriate and broad knowledge of the content, decent ability to apply knowledge, ability to present content in a structured way.• 26 - 27: Excellent - Precise and complete knowledge of the content, good ability to apply knowledge, strong analytical skills, clear and correct description.• 28 - 29: Outstanding - Extensive, complete, and in-depth knowledge of the content, good application of knowledge, strong analytical and synthesis skills, confident and correct description.• 30-30 with Honors - Very extensive, complete, and in-depth knowledge of the content, well-established ability to apply content, excellent analytical skills, synthesis, interdisciplinary connections, and mastery of description.
Further information	<p>Students are advised to rely exclusively on information/communications provided on the official websites of the Department of Computer Science or on social groups only if they are created and managed exclusively by the instructors of the respective courses:</p> <ul style="list-style-type: none">• https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-dilaurea• https://www.uniba.it/it/ricerca/dipartimenti/informatica• https://elearning.uniba.it/ <p>The syllabi for all courses are available at the following link:</p> <ul style="list-style-type: none">• https://elearning.uniba.it/ <p>Information that all students should know is written in the educational regulations of the Degree Programs available on the site:</p> <ul style="list-style-type: none">• https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-dilaurea <p>Students are advised to be wary of information and materials circulating on unofficial websites or social groups, as they are often unreliable, incorrect, or incomplete. For any doubts, request a meeting with the instructor according to the prescribed reception methods.</p> <p>Link to the course on the UNIBA e-learning platform: https://elearning.uniba.it/</p>