



Principali informazioni sull'insegnamento

Denominazione dell'insegnamento	Laboratorio di informatica (track cognomi A-L)	
Corso di studio	Informatica	
Anno Accademico	2023/24	
Crediti formativi universitari (CFU) / European Credit Transfer and Accumulation System (ECTS)	6 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	La frequenza è fortemente raccomandata	
Sito web del corso di studio	https://elearning.uniba.it/course/view.php?id=1962	


Docente/i

Nome e cognome	Fabio Abbattista
Indirizzo mail	Fabio.abbattista@uniba.it
Telefono	080-5443298
Sede	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.666, 6^ piano.
Sede virtuale	Microsoft Teams – Canale Laboratorio di informatica A – Informatica – 2022-23
Sito web del docente	
Ricevimento	Martedì 15:00 – 16:00, ma anche per e-mail

Syllabus

Obiettivi formativi	Il corso si propone di far acquisire le conoscenze necessarie per progettare, realizzare e testare software di media complessità, attraverso una progettazione modulare. Si approfondirà la programmazione imperativa strutturata per formulare soluzioni algoritmiche a problemi di complessità media. In particolare, lo studente estenderà la capacità di usare il linguaggio di programmazione C come strumento per modellare problemi e formalizzarne le soluzioni.		
Prerequisiti	Buona comprensione della lingua inglese		
Contenuti di insegnamento (Programma)	Mod	Argomenti (l'indicazione oraria si intende stimata)	Ore
	1	Ripasso di alcuni concetti di programmazione: Il problem solving La programmazione strutturata	3
	2	Ripasso di alcuni concetti di programmazione: Dati strutturati e dati utente Gestione dinamica della memoria	3
	3	Esercitazione di programmazione: Progettazione di un semplice gioco da tavolo	4
	4	Stile di Programmazione: La scelta dei nomi delle variabili e delle funzioni Indentazione e formato delle espressioni e istruzioni Uso dei Commenti Convenzioni di programmazione	3
	5	Esercitazione di programmazione: Realizzazione di un semplice gioco da tavolo	4
	6	Progettazione modulare: Modularizzazione e strutturazione dei programmi.	3



	Information Hiding Scope degli identificatori	
7	Esercitazione: Modularizzazione del progetto di un gioco da tavolo	4
8	Progettazione modulare: Moduli in linguaggio C Header file	3
9	Esercitazione: Modularizzazione del codice di un gioco da tavolo	4
10	Testing e Debugging: Tipi di test Lo unit test: white box e black box Strumenti e tecniche di debugging	3
11	Esercitazione: Modularizzazione del codice di un gioco da tavolo	4
12	Tecniche di testing: Test per classi di equivalenza Test per valori estremi	3
13	Esercitazione: Test del codice di un gioco da tavolo	4
14	Introduzione alla Complessità Computazionale	3
15	Analisi della complessità di alcuni algoritmi di ordinamento: bubble sort, selection sort, insertion sort, merge sort, quick sort	4
16	Realizzazione di un caso di studio: Definizione delle operazioni principali e dei dati strutturati del progetto Definizione dei moduli del progetto	4
17	Realizzazione di un caso di studio: Progettazione delle funzioni del progetto	4
18	Realizzazione di un caso di studio: Progettazione dei casi di test del progetto	4
19	Realizzazione di un caso di studio: Codifica e test del progetto	4
Testi di riferimento		
	<p>Testo da cui studiare:</p> <p>P. Deitel e H. Deitel Il linguaggio C – Fondamenti e tecniche di programmazione 8ª edizione - Pearson 2016 - ISBN: 9788891901651 (vanno bene anche le edizioni successive e precedenti dalla 4ª in poi)</p> <p>Testo integrativo, facoltativo:</p> <p>J.R. Hanly, E.B. Koffman, Problem solving e programmazione in C, Apogeo, 2013. ISBN: 8838786410</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 utilizzerà delle slide che verranno fornite</p> <p>Sul sito web del corso (v. sopra 'sito web del corso') sono disponibili:</p> <ul style="list-style-type: none"> ● Materiale didattico utilizzato a lezione; ● alcune tracce di prove scritte di esami, con esempi di tracce svolte; 	



Organizzazione della didattica			
Ore			
Totali	Didattica frontale	Laboratorio ed esercitazioni	Studio individuale
150 ore	24 ore	45 ore	81 ore
CFU/ETCS			
6 CFU	3 CFU	3 CFU	

Metodi didattici	
	Lezioni frontali, esercitazioni ed attività autonome e di gruppo in aula e a casa (come dettagliato nel programma). Gli studenti non frequentanti possono lavorare singolarmente prendendo accordi con il docente.

Risultati di apprendimento previsti	
Conoscenza e capacità di comprensione	<ul style="list-style-type: none">• Acquisire la capacità di analizzare e risolvere problemi di media complessità• Acquisire la capacità' di progettare e sviluppare programmi in un linguaggio di alto livello
Conoscenza e capacità di comprensione applicate	Lo studente dovrà acquisire competenze relative a: <ul style="list-style-type: none">• Traduzione di semplici algoritmi in programmi correttamente funzionanti e ben documentati;• Utilizzo di tecniche di programmazione difensiva, per limitare l'introduzione di malfunzionamenti nei programmi;• Verifica empirica della correttezza dei programmi mediante testing;• Capacità di problem-solving attraverso l'applicazione di nozioni apprese nelle discipline informatiche di base nella pratica della programmazione.
Competenze trasversali	<p>Autonomia di giudizio</p> <ul style="list-style-type: none">• Lo studente deve dimostrare di aver acquisito autonomia di giudizio e capacità di valutazione degli algoritmi sviluppati da lui o da terzi. <p>Abilità comunicative</p> <ul style="list-style-type: none">• Lo studente deve essere in grado di illustrare in modo appropriato le caratteristiche tecniche degli strumenti e delle metodologie informatiche apprese nel corso del primo anno di corso, relative allo sviluppo di programmi di media complessità. <p>Capacità di apprendere in modo autonomo</p> <ul style="list-style-type: none">• Lo studente dovrà mostrare di essere in grado di orientarsi agevolmente nelle problematiche relative alla comprensione e all'utilizzo delle tecnologie e dei metodi di competenza per lo sviluppo di algoritmi e per la loro traduzione in programmi per computer.

Valutazione	
--------------------	--



<p>Modalità di verifica dell'apprendimento</p>	<p>Realizzazione in itinere di un progetto, in gruppo. Il progetto e' lo stesso per tutti i gruppi. Il progetto deve essere completato entro la prima sessione di esami (Giugno/Luglio).</p> <p>Gli studenti che decidono di non partecipare al progetto durante il corso dovranno in ogni caso realizzare un progetto, in gruppo. Il progetto sara' diverso per ogni gruppo.</p> <p>Il voto finale conseguito viene proposto esclusivamente sulla piattaforma Esse3.</p> <p>Incentivi alla frequenza: l'eventuale lode viene più frequentemente attribuita agli studenti che per la stragrande maggioranza delle lezioni hanno frequentato, interagito nel corso della lezione, proposto soluzioni e risolto i casi proposti dal docente a lezione.</p>												
<p>Criteria di valutazione</p>	<p>Conoscenza e capacità di comprensione:</p> <ul style="list-style-type: none"> Lo studente dovrà dimostrare di aver acquisito conoscenze approfondite del linguaggio di programmazione C Lo studente dovrà dimostrare di aver acquisito capacità di progettazione e sviluppo di programmi di complessità elevata. Lo studente dovrà dimostrare di avere capacità di problem-solving anche per problemi complessi. <p>Conoscenza e capacità di comprensione applicate:</p> <ul style="list-style-type: none"> Lo studente dovrà dimostrare di saper produrre programmi (in C) funzionanti e di saperli documentare. Lo studente dovrà dimostrare di saper utilizzare tecniche di programmazione difensiva e di condurre una verifica empirica della correttezza dei programmi mediante tecniche di testing. Lo studente dovrà dimostrare di saper realizzare soluzioni efficienti <p>Autonomia di giudizio:</p> <ul style="list-style-type: none"> Abilità di valutazione della qualità di un approccio risolutivo adottato o adottabile per definire la soluzione ad un problema <p>Abilità comunicative:</p> <ul style="list-style-type: none"> Lo studente dovrà dimostrare di saper illustrare in modo appropriato la soluzione creata utilizzando un linguaggio tecnico corretto. <p>Capacità di apprendere:</p> <ul style="list-style-type: none"> Capacità di astrazione, di ragionamento per analogia e dimostrazione di creatività nella risoluzione dei problemi 												
<p>Criteria di misurazione dell'apprendimento e di attribuzione del voto finale</p>	<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> </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.
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.												



	<table border="1"><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></table>	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.
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/corsi-di-laurea• https://www.uniba.it/it/ricerca/dipartimenti/informatica• https://elearning.di.uniba.it/ <p>Le informazioni che tutti gli studenti dovrebbero conoscere sono scritte nei Regolamenti didattici e manifesti degli studi disponibili nel sito:</p> <ul style="list-style-type: none">• https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-di-laurea/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>				



Main information on the course

Course name	Programmazione (track cognomi A-L)	
Degree	Informatica	
Academic year	2023/24	
European Credit Transfer and Accumulation System (ECTS) , in Italian Crediti formativi universitari (CFU)	12 CFU	
Scientific Disciplinary Sector	ING-INF/05	
Course language	Italian	
Year of study	First	
Semester	1st semester, the exact dates are indicated annually in the degree course regulations	
Attendance requirement	It is highly recommended to attend classes	
Web site	https://groups.google.com/g/programmazione2013	

Teacher(s)	
Name and surname	Fabio Abbattista
e-mail	Fabio.abbattista@uniba.it
Phone	080-5443298
Office	Dipartimento di Informatica, Via Orabona 4, 70125, Bari. Stanza n.666, 6 ^a piano.
e-learning platform	Microsoft Teams – Canale Programmazione A – Informatica – 2022-23
Homepage	
Office hours	Tuesday 15:00 – 16:00, and by email

Syllabus

Course goals	<p>The course aims to introduce the basic elements to design, implement and test software of average complexity, by means of modular design.</p> <p>Students will learn more about structured imperative programming to formulate algorithmic solutions to problems of average complexity.</p> <p>In particular, the student will improve the ability to use the C programming language as a tool for modeling problems and formalizing their solutions.</p>																	
Prerequisites/requirements	Good understanding of the English language.																	
Course program	<table border="1"> <thead> <tr> <th>Mod</th> <th>Topics</th> <th>Hours</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>Presentazione del corso, contenuti, modalità d'esame, esoneri, frequenza alle lezioni, orari, modalità di esercitazione in aula, ecc. Introduzione alla programmazione Un primo algoritmo in pseudocodice: cucinare un primo</td> <td>3</td> </tr> <tr> <td>2</td> <td>Gli algoritmi Lo pseudocodice.</td> <td>3</td> </tr> <tr> <td>3</td> <td>Le strutture di controllo Il comando di selezione if Il comando di selezione if else. Il comando di iterazione while.</td> <td>3</td> </tr> <tr> <td>4</td> <td>Esercitazione sugli algoritmi: la somma di 2 numeri come incrementi successivi Esercitazione sugli algoritmi: il prodotto di 2 numeri come somme successive</td> <td>4</td> </tr> </tbody> </table>	Mod	Topics	Hours	1	Presentazione del corso, contenuti, modalità d'esame, esoneri, frequenza alle lezioni, orari, modalità di esercitazione in aula, ecc. Introduzione alla programmazione Un primo algoritmo in pseudocodice: cucinare un primo	3	2	Gli algoritmi Lo pseudocodice.	3	3	Le strutture di controllo Il comando di selezione if Il comando di selezione if else. Il comando di iterazione while.	3	4	Esercitazione sugli algoritmi: la somma di 2 numeri come incrementi successivi Esercitazione sugli algoritmi: il prodotto di 2 numeri come somme successive	4		
Mod	Topics	Hours																
1	Presentazione del corso, contenuti, modalità d'esame, esoneri, frequenza alle lezioni, orari, modalità di esercitazione in aula, ecc. Introduzione alla programmazione Un primo algoritmo in pseudocodice: cucinare un primo	3																
2	Gli algoritmi Lo pseudocodice.	3																
3	Le strutture di controllo Il comando di selezione if Il comando di selezione if else. Il comando di iterazione while.	3																
4	Esercitazione sugli algoritmi: la somma di 2 numeri come incrementi successivi Esercitazione sugli algoritmi: il prodotto di 2 numeri come somme successive	4																



5	Formulazione degli algoritmi con processo top down per raffinamenti successivi	3
6	Strutture di controllo nidificate	3
7	Esercitazione sulle strutture di controllo nidificate con uso di pseudocodice	4
8	Formulazione degli algoritmi con processo top down per raffinamenti successivi: studio di un caso, iterazione controllata da un valore sentinella.	3
9	Gli operatori logici Operatori di uguaglianza (=) e di assegnamento (=); Riassunto della programmazione strutturata.	3
10	Esercitazione sulle strutture di controllo nidificate con uso di pseudocodice	4
11	Tipi di dato semplici: interi, reali, caratteri e loro rappresentazione in C	3
12	I vettori La dichiarazione dei vettori Esempi sui vettori	3
13	Esercitazione sui vettori: ricerca del massimo	4
14	Studio di un caso: calcolare la media e la mediana usando i vettori La ricerca nei vettori (lineare e binaria)	3
15	Uso del compilatore Ambiente di sviluppo Eclipse Cenni al debugging Esercitazioni e approfondimenti degli argomenti: Un semplice programma C, visualizzare una riga di testo.	3
16	Esercitazione sui vettori: ricerca di un valore, ricerca di un valore vincolato	4
17	Istruzioni di controllo in C: selezione e iterazione	3
18	Istruzioni di Input/Output del C	3
19	Esercitazione in C: codifica degli algoritmi realizzati in precedenza	4
20	Astrazione sui dati. Tipi di dato utente Operazioni di accesso alle strutture dati utente	3
21	Matrici e tabelle come dati utente: i record	3
22	Manipolazione delle matrici Esercitazione sui vettori bidimensionali	4
23	Astrazione funzionale: funzioni e procedure e loro realizzazione in pseudocodice	3
24	Le funzioni in C: le definizioni di funzione e i prototipi di funzione	3
25	Esercitazione con le funzioni in C: un semplice programma strutturato in 3 funzioni, input, elaborazione e output	4
26	Lo stack delle chiamate di funzione e i record di attivazione Regole di visibilità in C	3
27	Invocare le funzioni: chiamata per valore e per riferimento	3
28	Esercitazione con le funzioni in C: lo scambio di 2 valori, somma di vettori	4
29	Vettori statici ed automatici Passare i vettori alle funzioni, passaggio per riferimento (indirizzo) e per valore	3
30	Puntatori: introduzione, dichiarazione e inizializzazione Operatore di indirizzo (&) e operatore di dereferenziazione (*) Puntatori come parametri di funzioni	3
31	Esercitazione sui puntatori: trovare il massimo e il minimo di un vettore usando il passaggio per riferimento	4
32	La gerarchia dei dati	3



		I file e gli stream Creare un file ad accesso sequenziale Leggere e scrivere i dati da un file ad accesso sequenziale	
	33	I file ad accesso casuale. Creare un file ad accesso casuale. Leggere e scrivere dati da un file ad accesso casuale.	3
	34	Esercitazione sui file sequenziali: creazione di un file di testo, modifica del contenuto e aggiornamento del file di testo	4
	35	Introduzione alla ricorsione ed esercizi: fattoriale ricorsivo, Fibonacci, Ricerca binaria ricorsiva e Torre di Hanoi	3
Books of reference	<p>Text to study from::</p> <p>P. Deitel e H. Deitel Il linguaggio C – Fondamenti e tecniche di programmazione 8ª edizione - Pearson 2016 - ISBN: 9788891901651</p> <p>Additional text, optional:</p> <p>J.R. Hanly, E.B. Koffman, Problem solving e programmazione in C, Apogeo, 2013. ISBN: 8838786410</p> <p>Students who wish can obtain texts on loan from the Library. Could it 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 the loan.</p>		
Note to the books	<p>During the lessons the teacher will use slides that retrace the contents of the book, therefore they will not be provided. The reference text contains all the topics of the course, therefore it is advisable to study from the text and to carry out independently and constantly all the exercises included at the end of each chapter covered in class.</p> <p>On the Google groups platform (see above 'virtual office') are available:</p> <ul style="list-style-type: none"> • supporting video material used in class; • some traces of written tests of exams, with examples of traces carried out; 		
Organization of the didactic activities			
Hours			
Total	Lectures	Practice sessions	Individual study
150 hours	24 hours	45 hours	81 hours
CFU/ETCS			
6 CFU	3 CFU	3 CFU	

Teaching methods	
	Lectures, exercises and autonomous and group activities in the classroom and at home (as detailed in the program). Non-attending students can work individually by making arrangements with the teacher.
Expected learning outcomes	
Knowledge and understanding	<ul style="list-style-type: none"> • Acquire skill to analyze and solve problems of average complexity; • Acquire the ability to design and implement programs with an high-level language programming;
Applying knowledge and understanding	Students should acquire the ability to:



	<ul style="list-style-type: none"> • Translate simple algorithms into well-documented and properly functioning programs; • Use defensive programming techniques to reduce the number of faults and bugs; • Verify the empirical correctness of programs; 																
Other skills	<p><i>Making judgements</i></p> <ul style="list-style-type: none"> • Acquire the ability to judge and evaluate the algorithms developed by himself or by third parties; <p><i>Communication</i></p> <ul style="list-style-type: none"> • Learn how to describe appropriately the technical characteristics of tools and methods concerning the software development, learned in the first academic year; <p><i>Learning skills</i></p> <ul style="list-style-type: none"> • Ability to easily understand how to use technologies and methods to develop algorithms and their translation in computer programs. 																
Assessment																	
Assessment methods	<p>To develop a software project as a group, in the course of the lessons. The project is the same for all the groups. The projects should be completed in the first exam session (June/July).</p> <p>Students who choose not to take part in the mid-term assessment tests, or fail to pass them, should develop a software project, as a group. The project will be different for each group.</p> <p>The final grade obtained is proposed exclusively on the Esse3 platform.</p> <p>Attendance incentives: any praise is most frequently given to students who, for the vast majority of lessons, have attended, interacted during the lesson, proposed solutions and solved the cases proposed by the teacher in class.</p>																
Evaluation criteria																	
Measurements and final grade	<table border="1"> <thead> <tr> <th>Mark</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>< 18 poor</td> <td>Fragmentary and superficial knowledge of the contents, errors in applying the concepts, deficient description.</td> </tr> <tr> <td>18 - 20</td> <td>Sufficient but general content knowledge, simple description, uncertainties in the application of theoretical concepts.</td> </tr> <tr> <td>21 - 23</td> <td>Appropriate but not in-depth knowledge of content, ability to apply theoretical concepts, ability to present content in a simple way.</td> </tr> <tr> <td>24 - 25</td> <td>Appropriate and extensive knowledge of the contents, good ability to apply knowledge, ability to present the contents in an articulated way.</td> </tr> <tr> <td>26 - 27</td> <td>Precise and complete content knowledge, good ability to apply knowledge, analytical skills, clear and correct description.</td> </tr> <tr> <td>28 - 29</td> <td>Wide, complete and in-depth knowledge of the contents, good application of the contents, good capacity for analysis and synthesis, safe and correct description.</td> </tr> <tr> <td>30 30 e lode</td> <td>Very broad, complete and in-depth knowledge of the contents, well-established ability to apply the contents, excellent capacity for analysis, synthesis and interdisciplinary connections, mastery of description.</td> </tr> </tbody> </table>	Mark	Description	< 18 poor	Fragmentary and superficial knowledge of the contents, errors in applying the concepts, deficient description.	18 - 20	Sufficient but general content knowledge, simple description, uncertainties in the application of theoretical concepts.	21 - 23	Appropriate but not in-depth knowledge of content, ability to apply theoretical concepts, ability to present content in a simple way.	24 - 25	Appropriate and extensive knowledge of the contents, good ability to apply knowledge, ability to present the contents in an articulated way.	26 - 27	Precise and complete content knowledge, good ability to apply knowledge, analytical skills, clear and correct description.	28 - 29	Wide, complete and in-depth knowledge of the contents, good application of the contents, good capacity for analysis and synthesis, safe and correct description.	30 30 e lode	Very broad, complete and in-depth knowledge of the contents, well-established ability to apply the contents, excellent capacity for analysis, synthesis and interdisciplinary connections, mastery of description.
Mark	Description																
< 18 poor	Fragmentary and superficial knowledge of the contents, errors in applying the concepts, deficient description.																
18 - 20	Sufficient but general content knowledge, simple description, uncertainties in the application of theoretical concepts.																
21 - 23	Appropriate but not in-depth knowledge of content, ability to apply theoretical concepts, ability to present content in a simple way.																
24 - 25	Appropriate and extensive knowledge of the contents, good ability to apply knowledge, ability to present the contents in an articulated way.																
26 - 27	Precise and complete content knowledge, good ability to apply knowledge, analytical skills, clear and correct description.																
28 - 29	Wide, complete and in-depth knowledge of the contents, good application of the contents, good capacity for analysis and synthesis, safe and correct description.																
30 30 e lode	Very broad, complete and in-depth knowledge of the contents, well-established ability to apply the contents, excellent capacity for analysis, synthesis and interdisciplinary connections, mastery of description.																
Further information	<p>Students are advised to rely exclusively on the information/communications provided on the official websites of the Computer Science Department, or on social groups only if set up and administered exclusively by the teachers of the related courses:</p> <ul style="list-style-type: none"> • https://www.uniba.it/it/ricerca/dipartimenti/informatica/didattica/corsi-dilaurea/ 																



corsi-di-laurea

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

Course schedules are available here:

- <https://programmi.di.uniba.it/>

The information that all students should know is written in the Teaching regulations and study posters available on the site:

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

Students are advised to be wary of information circulating on unofficial sites or social groups, as they are often found to be unreliable, incorrect or incomplete.

Link to the course on the Google groups platform:

<https://groups.google.com/g/programmazione2013>