Principali informazioni sull'insegnamento	
Titolo insegnamento	Programmazione [007398B]
Corso di studio	Informatica e Tecnologie per la Produzione del Software
Crediti formativi	9+3
Denominazione inglese	Computer Programming
Obbligo di frequenza	No, ma frequenza consigliata
Lingua di erogazione	Italiano

Docente responsabile	Nome Cognome	Indirizzo mail
	Giovanni Dimauro	giovanni.dimauro@uniba.it http://www.di.uniba.it/~dimauro/
Luogo e orario di ricevimento	Dipartimento Informatica	Martedì 11:00 – 13:00

Dettaglio credi formativi	Ambito disciplinare	SSD	Crediti
	Informatico, fisico, matematico, economico, linguistico	ING-INF/05-Sistemi di Elaborazione dell'Informazione	9+3

Modalità di erogazione	
Periodo di erogazione	1^ semestre
Anno di corso	1^
Modalità di erogazione	Lezioni frontali, esercitazioni

Organizzazione della didattica	
Ore totali	117 (corso) + 183 (studio individuale)
Ore di corso	72+45
Ore di studio individuale	153+30

Calendario	
Inizio attività didattiche	26 settembre 2022
Fine attività didattiche	13 gennaio 2022

Syllabus	
Prerequisiti	Buona comprensione della lingua inglese. Lettura individuale da parte dello studente del 1^ capitolo del testo di riferimento (v. sotto): "Introduzione ai computer, a Internet e al web"

Risultati di apprendimento previsti	• *Conoscenza e capacità di comprensione* Lo studente dovrà essere in grado di analizzare e risolvere semplici problemi, progettando e sviluppando programmi nel linguaggio C. • *Conoscenza e capacità di comprensione applicate* Lo studente dovrà acquisire competenze relative a: - Traduzione di semplici algoritmi in programmi correttamente funzionanti; - Verifica empirica della correttezza dei programmi mediante testing; - Capacità di individuazione di malfunzionamenti attraverso il debugging; - Capacità di problem-solving attraverso l'applicazione di nozioni apprese nelle discipline informatiche di base nella pratica della programmazione. • *Autonomia di giudizio* Lo studente deve dimostrare di aver acquisito autonomia di giudizio e di capacità di valutazione nell'ambito dello sviluppo di algoritmi. • *Capacità di apprendere* Lo studente dovrà mostrare di aver sviluppato capacità di apprendere e di orientarsi agilmente nelle problematiche relative alla comprensione e all'utilizzo delle tecnologie informatiche nel suo specifico campo di applicazione.

Contenuti di insegnamento

Mod	Argomenti	Ore
1	Presentazione del corso, contenuti, modalità d'esame, esoneri, frequenza alle lezioni, orari, modalità di esercitazione in aula, ecc. Risposte alle domande degli studenti. Introduzione alla programmazione Un semplice programma C: visualizzare una riga di testo. Un altro semplice programma C: sommare due interi.	3
2	Uso del compilatore ambiente di sviluppo Xcode cenni al debugging esercitazioni e approfondimenti degli argomenti: Un semplice programma C: visualizzare una riga di testo.	3
3	Nozioni sulla memoria L'aritmetica del C Prendere delle decisioni: gli operatori di uguaglianza e relazionali. Gli algoritmi Lo pseudocodice. Le strutture di controllo	3

	Il comando di selezione if	
4	Il comando di selezione if else. Il comando di iterazione while. Formulazione degli algoritmi: studio di un caso: l'iterazione controllata da un contatore. Formulazione degli algoritmi con processo top down per raffinamenti successivi: studio di un caso: iterazione controllata da un valore sentinella. Conversione di tipo, precisione.	4
5	Strutture di controllo nidificate esercitazione con uso di pseudocodice Formulazione degli algoritmi con processo top down per raffinamenti successivi: studio di un caso: strutture di controllo nidificate	3
6	Gli operatori di incremento e di decremento Gli elementi dell'iterazione Iterazione controllata da un contatore il comando di iterazione FOR note e osservazioni esempi di utilizzo del comando FOR Il comando di selezione multipla switch introduzione al comando break, studio di un caso. Il comando di iterazione do...while Le istruzioni break e continue	4
7	Gli operatori logici Operatori di uguaglianza (==) e di assegnamento (=); Riassunto della programmazione strutturata. I moduli di programma in C	3
8	Le funzioni della libreria matematica Le funzioni Le definizioni di funzione I prototipi di funzione Lo stack delle chiamate di funzione e i record di attivazione I file di intestazione Invocare le funzioni: chiamata per valore e per riferimento Generazione di numeri casuali	3
9	Esercitazione/autovalutazione: sviluppo flowchart, attività svolta con assistenza del docente	4
10	Esempio: un gioco d'azzardo Le classi di memoria Le regole di visibilità	4
11	La ricorsione	2
12	I vettori La dichiarazione dei vettori Esempi sui vettori	3
13	Esercitazione e studio individuale di materiale video prodotto dal Docente, a supporto delle lezioni teoriche. Il video descrive lo sviluppo di un flowchart prendendo spunto dall'esercizio del cap 4.38 del testo 'Il linguaggio C' di Deitel&Deitel. Il metodo di progettazione rispetta i principi della programmazione strutturata e utilizza strutture di controllo di base del linguaggio di programmazione C	4
14	Ulteriori esempi sui vettori	3

15	Vettori statici ed automatici Passare i vettori alle funzioni, passaggio per riferimento (indirizzo) e per valore, qualificatore 'const' Algoritmi di ordinamento: bubble sort	4
16	Studio di un caso: calcolare la media, mediana e la moda usando i vettori La ricerca nei vettori (lineare), algoritmo e programma per la ricerca lineare in un vettore	4
17	La ricerca nei vettori (binaria) Vettori multidimensionali.	4
18	Manipolazione delle matrici esercitazione sulle matrici bidimensionali (fig.6.22 del testo)	3
19	Esercitazione e studio individuale di materiale video prodotto dal Docente, a supporto delle lezioni teoriche. Il video descrive lo sviluppo di un flowchart prendendo spunto dall'esercizio 'dado a 20 facce' del testo 'Il linguaggio C' di Deitel&Deitel. Il metodo di progettazione rispetta i principi della programmazione strutturata e utilizza strutture di controllo di base del linguaggio di programmazione C	3
20	Esercitazione: Individuazione dei numeri primi. Crivello di Eratostene.	3
21	Esercitazione e studio individuale di materiale video prodotto dal docente, a supporto delle lezioni teoriche. Il video descrive il passaggio da un algoritmo descritto con flowchart al codice in Linguaggio C, prendendo spunto dall'esercizio 'dado a 20 facce' del testo 'Il linguaggio C' di Deitel&Deitel. Il metodo di progettazione rispetta i principi della programmazione strutturata e utilizza strutture di controllo di base del linguaggio di programmazione C	3
22	Esercitazione e studio individuale di materiale video prodotto dal Docente, a supporto delle lezioni teoriche. Il video descrive lo sviluppo di un flowchart prendendo spunto dall'esercizio 'Craps - un gioco d'azzardo' del testo 'Il linguaggio C' di Deitel&Deitel. Il metodo di progettazione rispetta i principi della programmazione strutturata e utilizza strutture di controllo di base del linguaggio di programmazione C	3
23	La gerarchia dei dati I file e gli stream Creare un file ad accesso sequenziale Leggere i dati da un file ad accesso sequenziale	4
24	Leggere i dati da un file ad accesso sequenziale (approfondimenti) Programma per l'interrogazione del credito.	3
25	I file ad accesso casuale Creare un file ad accesso casuale	4
26	Scrivere i dati in modo casuale in un file ad accesso casuale Leggere i dati in modo casuale da un file ad accesso casuale	3
27	Studio di un caso: un programma per l'elaborazione delle transazioni	3
28	Approfondimenti sulla ricorsione ed esercizi: fattoriale ricorsivo, Fibonacci, Ricerca binaria ricorsiva e Torre di Hanoi.	3
29	Puntatori: introduzione dichiarazione e inizializzazione operatore di indirizzo (&) e operatore di dereferenziazione (*) passaggio per valore e per riferimento puntatori come parametri di funzioni Esercizi: dichiarazione e stampa dei valori contenuti in variabili e puntatori, elevare al cubo una variabile (chiamata per valore e chiamata per riferimento), trovare il massimo e il minimo di un vettore usando il passaggio per riferimento.	3

30	Puntatori: puntatori const, operatori sui puntatori, esercizio bubble sort con puntatori, esercizio bubble sort in ordine crescente e decrescente usando puntatori a funzione, Array multidimensionali ed esercizio mescola carte	4
	Correzioni collettive esercitazione svolta. Attività didattica di preparazione all'esonero.	4
	simulazione autonoma in preparazione del 1^ esonero, assistita dal docente	3
	Auto correzione esonero assistita dal docente	4
	esercitazione autonoma, assistita dal docente; Correzione collettiva esercitazione svolta. Attività didattica di preparazione all'esonero.	3
	Esercitazione e Simulazione 2^ esonero	3

Programma	
Testo di riferimento 	P. Deitel e H. Deitel Il linguaggio C – Fondamenti e tecniche di programmazione 8^edizione Pearson 2016 ISBN: 9788891901651
Note ai testi di riferimento	Testo integrativo consigliato: Kim N. King, Programmazione in C, Maggioli Editore, ISBN 8838785821
Metodi didattici	Lezioni frontali ed esercitazioni in aula.
Metodi di valutazione	Una prova di esonero si tiene in prossimità della settimana di interruzione delle lezioni, normalmente collocata intorno alla metà di novembre. La prova consiste nella soluzione di un problema individuando l'algoritmo e sviluppando il relativo programma in C. Il voto ottenuto alla prova di esonero può essere speso esclusivamente nel primo

	appello utile. Si confida e si stimola quindi lo studio costante anche nel corso delle vacanze natalizie con lo scopo di porre lo studente in condizione di sostenere i 3 esami del primo semestre prima dell'inizio del secondo. Incentivi alla frequenza: L'eventuale lode potrà essere attribuita solo 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.
Criteri di valutazione	Lo studente dovrà dimostrare di aver acquisito la capacità di progettare un algoritmo per la soluzione di problemi con caratteristiche diverse. Inoltre deve aver sviluppato buone competenze nell'utilizzo del linguaggio C. L'esame finale consiste nella soluzione di un problema individuando l'algoritmo e sviluppando il relativo programma in C (per chi non ha sostenuto/superato la prova di esonero) e nella prova orale (oppure scritta in forma di test a risposta chiusa/aperta) utile a verificare la preparazione teorica
Altro	Gli studenti potranno unirsi al Forum del corso A.A. 2022/23 su Telegram, utilizzato per scopi didattici, al quale aderisce anche il docente: https://t.me/+V1eWiJ67GTBjNzNk

General information	
Academic subject	Computer Programming
Degree course	*Laurea degree*
Academic Year	*first*
European Credit Transfer and Accumulation System (ECTS)	9+3
Language	*Italian*
Academic calendar (starting and ending date)	*October 4, 2021 – January 14, 2022*
Attendance	*Not mandatory, but suggested*

Professor/ Lecturer	
Name and Surname	Giovanni Dimauro
E-mail	giovanni.dimauro@uniba.it
Telephone	
Department and address	*Dipartimento di Informatica, via Orabona 4 – Bari (ITALY)*
Virtual headquarters	
Tutoring (time and day)	Tuesday 11-13

Syllabus	
Learning Objectives	• *Knowledge and understanding* *The student should be able to analyze and solve simple problems, designing and developing programs in the C language.* • *Applied knowledge and understanding* *The student will have to acquire skills related to:* *- Translation of simple algorithms into properly functioning programs;*

	- Empirical verification of the correctness of the programs by testing; *- Ability to identify malfunctions through debugging;* *- Problem-solving skills through the application of notions learned in basic computer disciplines in the practice of programming.* *• Autonomy of judgment* *The student must demonstrate that he has acquired autonomy of judgment and evaluation skills in the context of the development of algorithms.* *• Ability to learn* *The student must show that they have developed the ability to learn and to orient themselves easily in the problems related to the understanding and use of information technologies in its specific field of application.*
Course prerequisites	*Good understanding of the English language. Individual reading by the student of the 1st chapter of the reference text (see below): "Introduction to computers, the Internet and the web"*
Contents	*Presentation of the course, contents, examination methods, exemptions, class attendance, timetables, classroom practice methods, etc.* *Answers to student questions.* *Introduction to programming* *A simple C program: display a line of text.* *Another simple C program: add two integers.* *Using the compiler* *Xcode development environment* *hints to debugging* *tutorials and in-depth analysis of the topics: A simple C program: display a line of text.* *Notions on memory* *The arithmetic of C* *Making decisions: equality and relational operators.*

	The algorithms *The pseudocode.* *The control structures* *The selection command if* *The if else selection command.* *The while iteration command.* *Formulation of algorithms: case study: iteration controlled by a counter.* *Formulation of algorithms with top down process for subsequent refinements: case study: iteration controlled by a sentinel value.* *Type conversion, precision.* *Nested control structures* *exercise with use of pseudocode* *Formulation of algorithms with top down process for subsequent refinements: case study: nested control structures* *The increase and decrease operators* *The elements of the iteration* *Iteration controlled by a counter* *the FOR iteration command* *notes and observations* *examples of using the FOR command* *The switch multiple selection command* *introduction to the break command, case study.* *The do ... while iteration command* *The break and continue statements* *The logical operators* *Equality (==) and assignment (=) operators;* *Summary of structured programming.* *The program modules in C*

	The functions of the math library *Functions* *Function definitions* *Function prototypes* *The function call stack and activation records* *The header files* *Invoke functions: call by value and by reference* *Generation of random numbers* *Exercise / self-assessment: flowchart development, activity carried out with the assistance of the teacher* *Example: a game of chance* *The memory classes* *The rules of visibility* *Recursion* *Carriers* *The declaration of the carriers* *Examples on vectors* *Exercise and individual study of video material produced by the teacher, to support the theoretical lessons. The video describes the development of a flowchart taking inspiration from the exercise of chapter 4.38 of the text 'The C language' by Deitel & Deitel. The design method respects the principles of structured programming and uses basic control structures of the C programming language* *More vector examples* *Static and automatic vectors* *Pass vectors to functions, pass by reference (address) and by value, 'const' qualifier* *Sorting algorithms: bubble sort* *Case study: calculating the mean, median and mode using vectors* *Search in vectors (linear), algorithm and program for linear search in a vector* *Search in vectors (binary)*

	Multidimensional vectors. *Manipulation of matrices* *exercise on two-dimensional matrices (fig.6.22 of the text)* *Exercise and individual study of video material produced by the teacher, to support the theoretical lessons. The video describes the development of a flowchart inspired by the '20-sided die' exercise of the text 'The C language' by Deitel & Deitel. The design method respects the principles of structured programming and uses basic control structures of the C programming language* *Exercise: Finding Prime Numbers. Sieve of Eratosthenes.* *Exercise and individual study of video material produced by the teacher, to support the theoretical lessons. The video describes the transition from an algorithm described with flowchart to the code in Language C, taking its cue from the exercise '20-sided die' of the text 'The C language' by Deitel & Deitel. The design method respects the principles of structured programming and uses basic control structures of the C programming language* *Exercise and individual study of video material produced by the teacher, to support the theoretical lessons. The video describes the development of a flowchart inspired by the exercise 'Craps - a game of chance' of the text 'The C language' by Deitel & Deitel. The design method respects the principles of structured programming and uses basic control structures of the C programming language* *The data hierarchy* *Files and streams* *Create a sequential access file* *Read data from a sequential access file* *Read data from a sequential access file (more details)* *Credit inquiry program.* *Random Access Files* *Create a random access file* *Write the data randomly to a random access file* *Read data randomly from a random access file* *Case Study: A Transaction Processing Program* *Insights on recursion and exercises: recursive factorial, Fibonacci, Recursive binary search and Tower of Hanoi.*

	Pointers: introduction *declaration and initialization* *address operator (&) and dereferencing operator (*)* *passage by value and by reference* *pointers as parameters of functions* *Exercises: declaring and printing the values contained in variables and pointers, cubeing a variable (called by value and called by reference), finding the maximum and minimum of a vector using passing by reference.* *Pointers:* *const pointers,* *pointer operators,* *bubble sort exercise with pointers,* *bubble sort exercise in ascending and descending order using function pointers,* *Multidimensional arrays and card shuffling exercise* *Collective corrections exercise carried out. Didactic activity in preparation for the exemption.* *autonomous simulation in preparation for the 1st exemption, assisted by the teacher* *Teacher-assisted self-correction* *independent exercise, assisted by the teacher; Collective correction of the exercise carried out. Didactic activity in preparation for the exemption.* *Exercise and Simulation 2nd exemption*
Books and bibliography	*P. Deitel e H. Deitel* *Il linguaggio C – Fondamenti e tecniche di programmazione* *8^edizione* *Pearson 2016* *ISBN: 9788891901651*
Additional materials	Kim N. King, Programmazione in C, Maggioli Editore, ISBN 8838785821

Work schedule			
Total	Lectures	Hands on (Laboratory, working groups, seminars, field trips)	Out-of-class study hours/ Self-study hours
Hours			
72		*45*	*183*
ECTS			
9		*3*	

Teaching strategy	*Learning by doing*
Expected learning outcomes	
Knowledge and understanding on:	○ The student should be able to analyze and solve simple problems, designing and developing programs in the C language.
Applying knowledge and understanding on:	• The student will have to acquire skills related to: ○ Translation of simple algorithms into properly functioning programs; ○ Empirical verification of the correctness of the programs through testing; ○ Ability to identify malfunctions through debugging; ○ Problem-solving skills through the application of notions learned in basic computer disciplines in the practice of programming.
Soft skills	• *Making informed judgments and choices* ○ The student must demonstrate that he has acquired autonomy of judgment and evaluation skills in the context of the development of algorithms. • *Communicating knowledge and understanding* ○ the student will have to demonstrate that they know how to disseminate the contents learned • *Capacities to continue learning* ○ The student must show that they have developed the ability to learn and to orient themselves easily in the problems related to the understanding and use of information technologies in its specific field of application.

Assessment and feedback	
Methods of assessment	*A test of exemption is held near the week of interruption of the lessons, usually placed around the middle of November. The test consists in solving a problem by identifying the algorithm and developing the relative program in C. The grade obtained in the exemption test can be spent*

	exclusively in the first useful session. Constant study is therefore trusted and stimulated even during the Christmas holidays with the aim of putting the student in a position to take the 3 exams of the first semester before the start of the second. *Attendance incentives: Any honors can be attributed only 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 during the lesson.*
Evaluation criteria	• *The student will have to demonstrate that he has acquired the ability to design an algorithm for solving problems with different characteristics. In addition, you must have developed good skills in the use of the C language.* • *The final exam consists in solving a problem by identifying the algorithm and developing the related program in C (for those who have not taken / passed the exemption test) and in the oral exam (or written in the form of a closed / open test.) useful to verify the theoretical preparation*
Criteria for assessment and attribution of the final mark	
Additional information	