



SYLLABUS A.A. 2019 /2020

PROGRAMMI DEGLI INSEGNAMENTI DEL I ANNO

SEM	Denominazione CORSO INTEGRATO / INSEGNAMENTO e CANALE DIDATTICO DI EROGAZIONE	Denominazione MODULO	S.S.D	AMBITO DISCIPLINARE/ TAF	CFU
Primo	ALGEBRA E GEOMETRIA - VARESE		MAT/02	A / Formazione matematico-fisica	9
	ALGEBRA E GEOMETRIA - COMO				
Primo	ARCHITETTURA DEGLI ELABORATORI - VARESE		ING- INF/05	A / Formazione informatica di base	9
	ARCHITETTURA DEGLI ELABORATORI - COMO				
Primo	PROGRAMMAZIONE - VARESE		INF/01	A / Formazione informatica di base	12
	PROGRAMMAZIONE - COMO				
Secondo	ANALISI MATEMATICA - VARESE		MAT/05	A / Formazione matematico-fisica	9
	ANALISI MATEMATICA - COMO				
Secondo	ALGORITMI E STRUTTURE DATI - VARESE		INF/01	B / Discipline Informatiche	9
	ALGORITMI E STRUTTURE DATI - COMO				
Secondo	INGLESE - VARESE		L-LIN/12	E / Per la conoscenza di almeno una lingua straniera	6
	INGLESE - COMO				
Secondo	LABORATORIO INTERDISCIPLINARE A e B	LABORATORIO INTERDISCIPLINARE A - VARESE	ING- INF/05	B / Discipline Informatiche	3 (di 6)
		LABORATORIO INTERDISCIPLINARE A - COMO			



ALGEBRA E GEOMETRIA – VARESE

Prof.ssa Brunella Gerla

Obiettivi formativi	Conoscenza e capacità di comprensione Il corso si propone di fornire le conoscenze di base di argomenti elementari di matematica discreta e di algebra lineare, in particolare sui seguenti argomenti: 1. Conoscere e comprendere i fondamenti del linguaggio matematico: Insiemi, funzioni, equazioni d'equivalenza 2. Conoscere le proprietà fondamentali dei numeri naturali e interi 3. Conoscere e comprendere le strutture algebriche, le sottostrutture e gli omomorfismi 4. Sapere utilizzare ed effettuare operazioni sulle matrici e conoscere gli elementi di base della geometria analitica per la rappresentazione nel piano cartesiano di oggetti geometrici 5. Saper risolvere i sistemi lineari con diversi metodi, conoscere e comprendere le caratteristiche dei sistemi lineari attraverso l'utilizzo delle matrici 6. Conoscere le definizioni fondamentali degli spazi vettoriali e delle applicazioni lineari. Tali conoscenze, oltre ad essere parte integrante del bagaglio culturale di uno studente di una laurea di carattere scientifico, sono rivolte a formare la capacità di astrazione dei problemi e delle informazioni attraverso la rappresentazione simbolica e matematica. Il corso affiancherà agli aspetti più teorici e metodologici della matematica, quegli aspetti più tecnici che permettono la risoluzione di esercizi e che rendono la matematica uno strumento di comprensione e di calcolo in vari settori applicativi. Conoscenza e capacità di comprensione applicate Durante il corso verrà data enfasi agli esempi legati ad applicazioni informatiche e soprattutto algoritmiche. In particolare si sottolineano aspetti relativi alla comprensione di proprietà di numeri naturali quali ricorsione e induzione. Una parte importante del corso sarà dedicata allo svolgimento di esercizi, sempre sottolineando che per riuscire a svolgere un esercizio c'è bisogno della totale comprensione dell'argomento trattato. Autonomia di giudizio e abilità comunicative I risultati di apprendimento attesi comprendono non solo la conoscenza dei termini e dei risultati tecnici, ma anche la capacità di saper affrontare una argomentazione matematica riuscendo a distinguere premesse e conclusioni. In questa ottica il linguaggio tecnico dello studente dovrà ampliarsi in modo da poter esprimere concetti matematici astratti. Capacità di apprendere Durante il corso verrà sottolineata l'importanza di un metodo di studio appropriato, in particolare cercando di favorire uno studio critico (come per esempio chiedersi sempre il perché di certe affermazioni matematiche), in modo da rendere autoevidenti allo studente le proprie lacune.
Prerequisiti	Non sono richieste conoscenze matematiche specifiche oltre quelle fornite da una qualsiasi scuola superiore.
Contenuti	L'acquisizione delle diverse conoscenze ed abilità attese si svilupperà in modo parallelo lungo tutto l'insegnamento, in cui verranno trattati i seguenti argomenti (divisi per obiettivi formativi): • Basi di matematica discreta (obiettivo 1): Teoremi e metodi di dimostrazione:

implicazione, contronominale, dimostrazioni per assurdo. Quantificatori e negazione. Principio di induzione, esempi e esercizi. (4h)

Insiemi, elementi di un insieme, appartenenza e inclusione, sottoinsiemi, insieme delle parti di un insieme, cardinalità di un insieme finito, diagrammi di Venn. Operazioni tra insiemi: unione, intersezione e complemento. Coppie e prodotto cartesiano. Contare gli elementi degli insiemi finiti. (4h)

Relazioni, relazioni binarie, proprietà riflessiva, simmetrica e transitiva. Relazioni d'equivalenza, classe d'equivalenza, insieme quoziente. Partizioni, teorema fondamentale delle relazioni d'equivalenza. Relazioni d'ordine, esempi: divisibilità tra numeri interi, prefissi, inclusioni tra insiemi. Elementi non confrontabili. Massimo e minimo. Estremo inferiore e estremo superiore. (6h)

Funzioni, dominio e codominio, immagine e preimmagine. Funzioni iniettive, suriettive e biiettive. Inversa di una funzione biettiva. Composizione di funzioni. (4h)

- Proprietà aritmetiche (obiettivo formativo 2):

Elementi di calcolo combinatorio: disposizioni semplici e con ripetizione, fattoriale, coefficiente binomiale. Disposizioni e combinazioni, contare le funzioni e le funzioni iniettive. (6h)

Algoritmo euclideo delle divisioni successive per il calcolo del MCD, numeri primi, teorema fondamentale dell'aritmetica (con dimostrazione), teorema sull'esistenza di infiniti numeri primi (con dimostrazione). (4h)

Numerazione in base n . Relazione di congruenza modulo n . Risolvere le congruenze lineari. Insieme delle classi di resto modulo n . (6h)

Operazioni su un insieme, proprietà commutativa e associativa, elemento neutro e elementi invertibili. Elementi invertibili in Z_m , funzione di Eulero. (4h)

- Strutture algebriche (obiettivo formativo 3)

Monoidi e gruppi. Esempi numerici e non (monoide delle parole, gruppo delle permutazioni). Definizione di sottogruppo, esempi. Esempio del gruppo delle matrici quadrate di ordine 2, con determinante diverso da zero. Sottogruppi, relazione d'equivalenza determinata da un sottogruppo, laterale destro, teorema di Lagrange. (4h)

Anelli, esempi (anello degli interi modulo n , anello delle matrici su R , anello dei polinomi). Elementi divisori dello 0 e invertibili. Campi, esempi (campo dei reali, campo dei complessi). (4h)

- Matrici (obiettivo formativo 4)

Matrici su un campo, operazioni tra matrici. Determinante e Rango (metodo di Laplace e Sarrus, metodo di Kronecker per il rango). Inversa di una matrice. Riduzione in forma triangolare. (4h)

- Sistemi lineari (obiettivo formativo 5)

Sistemi di equazioni lineari (omogenei e non omogenei): Metodo di Gauss-Jordan. Teorema di Rouchè-Capelli, Teorema di Cramer. (8h)

- Spazi vettoriali (obiettivo formativo 6)

Definizione di spazio vettoriale ed esempi. Sottospazi vettoriali. Insieme linearmente indipendente. Sottospazi generati. Spazio delle soluzioni di un sistema omogeneo. Basi e dimensione di uno spazio vettoriale. Dimensione di un sottospazio. (4h)

Applicazioni lineari: matrice associata ad una applicazione lineare. Nucleo e immagine di una applicazione lineare. Teorema di nullità più rango. (6h)

Autovalori e autovettori, molteplicità geometrica e algebrica di un autovalore, basi formate da autovettori. (4h)

Metodi didattici	Il corso prevede lezioni frontali (72 h) Le lezioni sono dedicate ad illustrare i concetti base della matematica discreta e dell'algebra lineare e sono accompagnate da esempi ed esercizi che aiuteranno nella comprensione degli argomenti.
Modalità di verifica dell'apprendimento	L'esame consiste in una prova scritta centrata sulla risoluzione degli esercizi relativi agli argomenti trattati nel corso e ad alcune domande di teoria, seguita da una prova orale facoltativa. Il voto massimo che si può avere con l'esame scritto è 28. Possono sostenere la prova orale facoltativa gli studenti che abbiano ottenuto almeno 25 allo scritto. Esempi di prove scritte passate si trovano sul sito di e-learning del corso.
Testi di riferimento	Saranno fornite delle dispense sul sito di elearning. Altri testi consigliati sono: • Introduzione alla Matematica Discreta, di M Bianchi e A. Gillio, McGraw-Hill. • Elementi di Matematica Discreta e Algebra Lineare di F. Dalla Volta e M. Rigoli, Pearson Education, 2007.
Altre informazioni	La docente riceve per appuntamento, previa richiesta via e-mail a brunella.gerla@uninsubria.it

ALGEBRA E GEOMETRIA – COMO

Prof. Stefano Serra Capizzano

Obiettivi formativi	Conoscenza e capacità di comprensione Il corso si propone di fornire le conoscenze di base di argomenti elementari di matematica discreta, come insiemi, funzioni, equazioni d'equivalenza e strutture algebriche, e di algebra lineare come sistemi lineari e matrici e cenni di geometria analitica. Tali conoscenze, oltre ad essere parte integrante del bagaglio culturale di uno studente di una laurea di carattere scientifico, sono rivolte a formare la capacità di astrazione dei problemi e delle informazioni attraverso la rappresentazione simbolica e matematica. Il corso affiancherà agli aspetti più teorici e metodologici della matematica, quegli aspetti più tecnici che permettono la risoluzione di esercizi e che rendono la matematica uno strumento di comprensione e di calcolo in vari settori applicativi. Conoscenza e capacità di comprensione applicate Durante il corso verrà data enfasi agli esempi legati ad applicazioni informatiche e soprattutto algoritmiche. In particolare si sottolineano aspetti relativi alla comprensione di proprietà di numeri naturali quali ricorsione e induzione. Una parte importante del corso sarà dedicata allo svolgimento di esercizi, sempre sottolineando che per riuscire a svolgere un esercizio c'è bisogno della totale comprensione dell'argomento trattato. Autonomia di giudizio e abilità comunicative I risultati di apprendimento attesi comprendono non solo la conoscenza dei termini e dei risultati tecnici, ma anche la capacità di saper affrontare una argomentazione matematica riuscendo a distinguere premesse e conclusioni. In questa ottica il linguaggio tecnico dello studente dovrà ampliarsi in modo da poter esprimere concetti matematici astratti. Capacità di apprendere Durante il corso verrà sottolineata l'importanza di un metodo di studio appropriato, in particolare cercando di favorire uno studio critico (come per esempio chiedersi sempre
---------------------	---

	il perché di certe affermazioni matematiche), in modo da rendere autoevidenti allo studente le proprie lacune.
Prerequisiti	Non sono richieste conoscenze matematiche specifiche oltre quelle fornite da una qualsiasi scuola superiore.
Contenuti	L'acquisizione delle diverse conoscenze ed abilità attese si svilupperà in modo parallelo lungo tutto l'insegnamento, in cui verranno trattati i seguenti argomenti: Teoremi e metodi di dimostrazione: implicazione, contronominale, dimostrazioni per assurdo. Quantificatori e negazione. Principio di induzione, esempi e esercizi. (4h) Insiemi, elementi di un insieme, appartenenza e inclusione, sottoinsiemi, insieme delle parti di un insieme, cardinalità di un insieme finito, diagrammi di Venn. Operazioni tra insiemi: unione, intersezione e complemento. Coppie e prodotto cartesiano. Contare gli elementi degli insiemi finiti. (4h) Relazioni, relazioni binarie, proprietà riflessiva, simmetrica e transitiva. Relazioni d'equivalenza, classe d'equivalenza, insieme quoziente. Partizioni, teorema fondamentale delle relazioni d'equivalenza. Relazioni d'ordine, esempi: divisibilità tra numeri interi, prefissi, inclusioni tra insiemi. Elementi non confrontabili. Massimo e minimo. Estremo inferiore e estremo superiore. (6h) Funzioni, dominio e codominio, immagine e preimmagine. Funzioni iniettive, suriettive e biettive. Inversa di una funzione biettiva. Composizione di funzioni. (4h) Elementi di calcolo combinatorio: disposizioni semplici e con ripetizione, fattoriale, coefficiente binomiale. Disposizioni e combinazioni, contare le funzioni e le funzioni iniettive. (6h) Algoritmo euclideo delle divisioni successive per il calcolo del MCD, numeri primi, teorema fondamentale dell'aritmetica (con dimostrazione), teorema sull'esistenza di infiniti numeri primi (con dimostrazione). (4h) Numerazione in base n. Relazione di congruenza modulo n. Risolvere le congruenze lineari. Insieme delle classi di resto modulo n. (6h) Operazioni su un insieme, proprietà commutativa e associativa, elemento neutro e elementi invertibili. Elementi invertibili in Z_m, funzione di Eulero. (4h) Monoidi e gruppi. Esempi numerici e non (monoide delle parole, gruppo delle permutazioni). Definizione di sottogruppo, esempi. Esempio del gruppo delle matrici quadrate di ordine 2, con determinante diverso da zero. Sottogruppi, relazione d'equivalenza determinata da un sottogruppo, laterale destro, teorema di Lagrange. (4h) Anelli, esempi (anello degli interi modulo n, anello delle matrici su R, anello dei polinomi). Elementi divisori dello 0 e invertibili. Campi, esempi (campo dei reali, campo dei complessi). (4h) Matrici su un campo, operazioni tra matrici. Determinante e Rango (metodo di Laplace e Sarrus, metodo di Kronecker per il rango). Inversa di una matrice. Riduzione in forma triangolare. (4h) Sistemi di equazioni lineari (omogenei e non omogenei): Metodo di Gauss-Jordan. Teorema di Rouchè-Capelli, Teorema di Cramer. (8h) Definizione di spazio vettoriale ed esempi. Sottospazi vettoriali. Insieme linearmente indipendente. Sottospazi generati. Spazio delle soluzioni di un sistema omogeneo. Basi e dimensione di uno spazio vettoriale. Dimensione di un sottospazio. (4h) Applicazioni lineari: matrice associata ad una applicazione lineare. Nucleo e immagine di una applicazione lineare. Teorema di nullità più rango. (6h) Autovalori e autovettori, molteplicità geometrica e algebrica di un autovalore, basi

	formate da autovettori. (4h)
Metodi didattici	Lezioni frontali e esercitazioni
Modalità di verifica dell'apprendimento	L'esame consiste in una prova scritta centrata sulla risoluzione degli esercizi relativi agli argomenti trattati nel corso e ad alcune domande di teoria, seguita da una prova orale facoltativa. Il voto massimo che si può avere con l'esame scritto è 28. Possono sostenere la prova orale facoltativa gli studenti che abbiano ottenuto almeno 25 allo scritto. Esempi di prove scritte passate si trovano sul sito di e-learning del corso.
Testi di riferimento	Saranno fornite delle dispense sul sito di elearning. Altri testi consigliati sono: • Introduzione alla Matematica Discreta, di M Bianchi e A. Gillio, McGraw-Hill. • Elementi di Matematica Discreta e Algebra Lineare di F. Dalla Volta e M. Rigoli, Pearson Education, 2007.
Altre informazioni	Ricevimento studenti previo appuntamento per e-mail stefano.serracapizzano@uninsubria.it

ARCHITETTURA DEGLI ELABORATORI – VARESE

Dott. Davide Tosi

Obiettivi formativi	Il corso ha lo scopo di rendere gli studenti capaci di comprendere il progetto e la programmazione degli elementi costruttivi degli elaboratori. Gli argomenti in programma sono selezionati sulla base di una consolidata tradizione nell'insegnamento dell'argomento, basata sulla visione della macchina come stratificazione di macchine reali o virtuali. Il corso mira a fornire tutte le competenze relative: 1. ai fondamenti dei circuiti elettronici (visti a livello logico) e le capacità fondamentali di progettazione; 2. all'organizzazione dell'hardware degli elaboratori; 3. alle funzioni svolte dall'hardware e utilizzate dai sistemi operativi e dai programmi applicativi a livello di memoria; 4. alla progettazione di un processore reale (macchina MIPS); 5. alla programmazione in linguaggio Assembly; Lo studente acquisisce quindi conoscenza e capacità di comprensione in riferimento: • ai principi costruttivi e delle caratteristiche delle memorie elettroniche comunemente usate nei calcolatori; • ai principi costruttivi e delle caratteristiche funzionali delle memorie cache; • ai principi costruttivi delle unità di elaborazione (datapath) e delle unità di controllo; • alla capacità di progettare unità di controllo mediante microprogrammazione; • ai principi costruttivi e delle modalità di funzionamento dei bus di connessione tra elementi dell'elaboratore; • all'organizzazione e delle modalità d'uso e di programmazione dei dispositivi di input/output; Le abilità e capacità acquisite dallo studente al termine del corso sono: • capacità di analizzare, progettare, ottimizzare circuiti combinatori, sia mediante tecniche basate sulla conoscenza dell'algebra Booleana sia utilizzando componenti di libreria comunemente disponibili sul mercato; • capacità di valutare le prestazioni dei circuiti;
---------------------	---

	• capacità di analizzare, progettare, ottimizzare circuiti sequenziali, eventualmente utilizzando componenti di libreria comunemente disponibili sul mercato; • capacità di valutare le prestazioni delle memorie cache; • abilità di scrivere programmi, anche complessi, in linguaggio di programmazione assembly; • abilità di valutare autonomamente le esigenze contrastanti che si presentano nella progettazione di circuiti (costo di progettazione, costo dei componenti, velocità di elaborazione) e prendere decisioni in merito. Grazie alla capacità –sviluppata durante tutto il corso– di riportare puntualmente tecniche, metodi e strumenti alle caratteristiche dell'organizzazione e della progettazione degli elaboratori e delle loro componenti, lo studente svilupperà la capacità di apprendere nuove tecniche, metodi e strumenti, e anche ad affinare ed adattare autonomamente quanto appreso nel corso. Inoltre, lo studente saprà usare i formalismi normalmente usati per descrivere circuiti e parti dei calcolatori e conoscerà la terminologia tecnica inglese comunemente usata nel campo.
Prerequisiti	Non è richiesta alcuna conoscenza particolare di tipo tecnico scientifico se non una buona preparazione matematico/fisica acquisita durante la scuola secondaria di secondo grado. È opportuna una buona comprensione della lingua inglese scritta, che permetta l'accesso alla grande quantità di materiale didattico, pubblicazioni, manuali, programmi, ecc. disponibili sull'argomento.
Contenuti	Le lezioni affronteranno i seguenti argomenti: • L'architettura strutturata (a "layer") dei calcolatori (14h + 4h es., obiettivo formativo 1) - o Principi ed elementi costitutivi di un tipico calcolatore elettronico - o Rappresentazione binaria dei numeri, dei caratteri e di altre informazioni - o Operatori e proprietà dell'algebra di Boole. Corrispondenza tra operatori Booleani e porte logiche • Circuiti combinatori e Sequenziali (18h + 6h es., obiettivo formativo 2) - o Sintesi di reti combinatorie, forme normali, mappe di Karnaugh - o Librerie di circuiti combinatori, ALU, sintesi di circuiti combinatori mediante elementi di libreria - o Bistabili, elementi sequenziali di libreria (vari tipi di registri). Progettazione di circuiti sequenziali • Studio e Progettazione delle Memorie di un Calcolatore (8h + 4h es., obiettivo formativo 3) - o Principio di località e gerarchie di memoria (memorie cache) - o Memoria virtuale, paginazione • Costruzione di un Data Path (8h + 2h es., obiettivo formativo 4) - o CPU a ciclo singolo e suo controllo - o CPU multiciclo - o Unità di controllo cablata e microprogrammata. Implementazione di istruzioni mediante microprogrammazione - o Architetture a pipeline (cenni) - o I/O (programmato, a interrupt, DMA), BUS • Struttura e istruzioni del linguaggio macchina (8h + 8h es., obiettivo formativo 5) - o Instruction Set Architecture della macchina MIPS

	o Programmazione in assembly MIPS: uso delle istruzioni, procedure; uso della pila; record di attivazione; ricorsione
Metodi didattici	Il corso si articola in lezioni frontali (56 ore) ed esercitazioni (24 ore) ripartite come dettagliato nella sezione Contenuti del Corso. Le lezioni frontali devono dare agli studenti tutti gli strumenti per poi comprendere ed applicare in contesti pratici e reali gli aspetti teorici appresi. Le esercitazioni permettono invece di applicare gli aspetti teorici a casi reali di progettazione. Ampia autonomia di risoluzione dei problemi viene lasciata agli studenti durante le ore di esercitazione. Alcune esercitazioni sono svolte con il supporto di strumenti informatici, sia di uso generale (come i fogli di calcolo) sia specifici (mappe di Karnaugh, simulatori di data path ecc.). Le lezioni sulla programmazione assembly vengono svolte con l'ausilio di uno strumento di programmazione e di simulazione del comportamento della macchina MIPS. L'impegno personale di rielaborazione autonoma da parte dello studente è tarato sul valore canonico di 25 ore complessive per CFU.
Modalità di verifica dell'apprendimento	L'obiettivo della prova d'esame è l'accertamento dell'acquisizione delle conoscenze e delle abilità descritte nella sezione "Obiettivi del corso", valutando il livello di conoscenza e soprattutto la capacità di mettere in pratica, anche integrandole tra loro, le tecniche e contenuti visti a lezione. L'apprendimento è verificato mediante una prova scritta da svolgersi in aula tradizionale della durata indicativa di due ore, senza l'aiuto di appunti o libri. La prova scritta consiste nello svolgimento di 4/5 esercizi, dei quali 3/4 richiedono capacità progettuali, e uno serve invece a verificare conoscenze acquisite. Ogni esercizio si riferisce ad uno specifico argomento e obiettivo formativo. La prova orale viene sostenuta solo su richiesta del docente nel caso lo scritto lasci dei dubbi sulla preparazione dello studente, e comporta una variazione massima di tre punti del punteggio acquisito nello scritto. In ogni caso, lo studente può prendere visione della correzione del proprio elaborato e discuterne la valutazione. Questo può talvolta portare a una revisione del punteggio. L'eventuale prova orale verte sull'accertamento della capacità di sintesi delle conoscenze acquisite, con particolare riguardo alla capacità di identificare gli elementi di teoria da utilizzare in attività di progettazione o programmazione. La conoscenza della terminologia specifica di dominio viene testata implicitamente, poiché domande e specifiche dei problemi utilizzano tale terminologia. Sono previste altresì due prove parziali intermedie oltre agli appelli generali. Il voto è espresso in trentesimi.
Testi di riferimento	I testi consigliati sono: - o Patterson & Hennessy, "Computer organization and design – The hardware/software interface", Elsevier - o Bolchini & Sciuto, "Informatica 2 sistemi digitali", Progetto Leonardo, Soc. Editrice Esculapio Bologna Le slide delle lezioni in formato PDF sono messe a disposizione sulla piattaforma e-learning di Ateneo ove sono disponibili anche il testo dei problemi visti durante le esercitazioni e le soluzioni proposte. Per consentire agli studenti di prepararsi adeguatamente agli esami e di verificare la propria preparazione, sul sito dell'e-learning di Ateneo sono anche disponibili i temi d'esame dal 2006 a oggi, e le relative tracce di soluzione.

Altre informazioni	Il docente riceve su appuntamento, previa richiesta via e-mail a davide.tosi@uninsubria.it . Il docente risponde solo alle e-mail firmate e provenienti dal dominio studenti.uninsubria.it.
--------------------	--

ARCHITETTURA DEGLI ELABORATORI – COMO

Dott. Sergio Moriani

Obiettivi formativi	Il corso ha lo scopo di rendere gli studenti capaci di comprendere il progetto e la programmazione degli elementi costruttivi degli elaboratori. Gli argomenti in programma sono selezionati sulla base di una consolidata tradizione nell'insegnamento dell'argomento, basata sulla visione della macchina come stratificazione di macchine reali o virtuali. Il corso mira a fornire tutte le competenze relative: 1. ai fondamenti dei circuiti elettronici (visti a livello logico) e le capacità fondamentali di progettazione; 2. all'organizzazione dell'hardware degli elaboratori; 3. alle funzioni svolte dall'hardware e utilizzate dai sistemi operativi e dai programmi applicativi a livello di memoria; 4. alla progettazione di un processore reale (macchina MIPS); 5. alla programmazione in linguaggio Assembly; Lo studente acquisisce quindi conoscenza e capacità di comprensione in riferimento: • ai principi costruttivi e delle caratteristiche delle memorie elettroniche comunemente usate nei calcolatori; • ai principi costruttivi e delle caratteristiche funzionali delle memorie cache; • ai principi costruttivi delle unità di elaborazione (datapath) e delle unità di controllo; • alla capacità di progettare unità di controllo mediante microprogrammazione; • ai principi costruttivi e delle modalità di funzionamento dei bus di connessione tra elementi dell'elaboratore; • all'organizzazione e delle modalità d'uso e di programmazione dei dispositivi di input/output; Le abilità e capacità acquisite dallo studente al termine del corso sono: • capacità di analizzare, progettare, ottimizzare circuiti combinatori, sia mediante tecniche basate sulla conoscenza dell'algebra Booleana sia utilizzando componenti di libreria comunemente disponibili sul mercato; • capacità di valutare le prestazioni dei circuiti; • capacità di analizzare, progettare, ottimizzare circuiti sequenziali, eventualmente utilizzando componenti di libreria comunemente disponibili sul mercato; • capacità di valutare le prestazioni delle memorie cache; • abilità di scrivere programmi, anche complessi, in linguaggio di programmazione assembly; • abilità di valutare autonomamente le esigenze contrastanti che si presentano nella progettazione di circuiti (costo di progettazione, costo dei componenti, velocità di elaborazione) e prendere decisioni in merito. Grazie alla capacità –sviluppata durante tutto il corso– di riportare puntualmente tecniche, metodi e strumenti alle caratteristiche dell'organizzazione e della progettazione degli elaboratori e delle loro componenti, lo studente svilupperà la capacità di apprendere nuove tecniche, metodi e strumenti, e anche ad affinare ed
---------------------	---

	adattare autonomamente quanto appreso nel corso. Inoltre, lo studente saprà usare i formalismi normalmente usati per descrivere circuiti e parti dei calcolatori e conoscerà la terminologia tecnica inglese comunemente usata nel campo.
Prerequisiti	Non è richiesta alcuna conoscenza particolare di tipo tecnico scientifico se non una buona preparazione matematico/fisica acquisita durante la scuola secondaria di secondo grado. È opportuna una buona comprensione della lingua inglese scritta, che permetta l'accesso alla grande quantità di materiale didattico, pubblicazioni, manuali, programmi, ecc. disponibili sull'argomento.
Contenuti	Le lezioni affronteranno i seguenti argomenti: • L'architettura strutturata (a "layer") dei calcolatori (14h + 4h es., obiettivo formativo 1) - o Principi ed elementi costitutivi di un tipico calcolatore elettronico - o Rappresentazione binaria dei numeri, dei caratteri e di altre informazioni - o Operatori e proprietà dell'algebra di Boole. Corrispondenza tra operatori Booleani e porte logiche • Circuiti combinatori e Sequenziali (18h + 6h es., obiettivo formativo 2) - o Sintesi di reti combinatorie, forme normali, mappe di Karnaugh - o Librerie di circuiti combinatori, ALU, sintesi di circuiti combinatori mediante elementi di libreria - o Bistabili, elementi sequenziali di libreria (vari tipi di registri). Progettazione di circuiti sequenziali • Studio e Progettazione delle Memorie di un Calcolatore (8h + 4h es., obiettivo formativo 3) - o Principio di località e gerarchie di memoria (memorie cache) - o Memoria virtuale, paginazione • Costruzione di un Data Path (8h + 2h es., obiettivo formativo 4) - o CPU a ciclo singolo e suo controllo - o CPU multiciclo - o Unità di controllo cablata e microprogrammata. Implementazione di istruzioni mediante microprogrammazione - o Architetture a pipeline (cenni) - o I/O (programmato, a interrupt, DMA), BUS • Struttura e istruzioni del linguaggio macchina (8h + 8h es., obiettivo formativo 5) - o Instruction Set Architecture della macchina MIPS - o Programmazione in assembly MIPS: uso delle istruzioni, procedure; uso della pila; record di attivazione; ricorsione
Metodi didattici	Da definire
Modalità di verifica dell'apprendimento	Da definire
Testi di riferimento	Da definire
Altre informazioni	Da definire

Obiettivi formativi	Questo corso fornisce un'introduzione alla programmazione dei calcolatori mediante il linguaggio Java. L'obiettivo primario è quello di fornire agli studenti le conoscenze necessarie alla comprensione e all'utilizzo del linguaggio Java e all'applicazione del linguaggio alla risoluzione di problemi. Il corso affianca quindi all'insegnamento teorico del linguaggio Java e delle sue strutture, esercitazioni e laboratori. Al termine di questo corso lo studente: 1. Conosce i principali elementi che caratterizzano un linguaggio di programmazione (sintassi, semantica, meccanismi implementativi); 2. Conosce le strutture di controllo, le strutture dati di base e le astrazioni principali fornite dal linguaggio Java e in generale dai linguaggi ad oggetti; 3. Possiede le conoscenze necessarie per affrontare individualmente lo studio di un linguaggio di programmazione e per comprendere i problemi relativi allo sviluppo e all'applicazione dei linguaggi di programmazione. 4. È in grado di applicare il linguaggio di programmazione alla risoluzione di semplici problemi individuando le astrazioni più adatte alla formalizzazione del problema in un linguaggio ad oggetti. 5. È in grado di individuare lo schema concettuale di risoluzione di un problema e di utilizzare il linguaggio di programmazione per l'organizzazione della sua soluzione. 6. È in grado di comunicare in modo compiuto e con la corretta terminologia le caratteristiche di un linguaggio di programmazione e la struttura delle applicazioni realizzate.
Prerequisiti	Non sono richieste conoscenze informatiche/matematiche specifiche oltre quelle fornite da una qualsiasi scuola superiore.
Contenuti	L'acquisizione delle diverse conoscenze ed abilità attese si svilupperà in modo parallelo lungo tutto l'insegnamento, in cui verranno trattati i seguenti argomenti: - Introduzione ai linguaggi di programmazione: astrazioni, algoritmi, sintassi, semantica e strumenti di sviluppo (lezioni 3 h, obiettivo formativo 1); - Introduzione al linguaggio Java. Concetti di base, compilazione, esecuzione, struttura applicazioni (lezioni 3 h, obiettivi formativi 1,2); - Tipi primitivi ed espressioni (lezioni 8 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 1, 2, 3); - Strutture di controllo (lezioni 8 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 1, 2, 3); - Array (lezioni 4 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4); - Gerarchia dei tipi riferimento, cast e polimorfismo (lezioni 6 h, esercitazioni 1 h, laboratorio 4 h, obiettivi formativi 2, 3, 4, 5, 6); - Implementazione delle classi (lezioni 8 h, esercitazioni 1 h, laboratorio ore 4 h, obiettivi formativi 2, 3, 4, 5, 6); - Implementazione dell'ereditarietà (lezioni 8 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Interfacce (lezioni 2 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Variabili, scope e adombramento (lezioni ore 2; esercitazioni 1 h; laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Early- e late-binding in Java (lezioni 4 h esercitazioni 1 h, laboratorio 2 h, obiettivi

	formativi 2, 3, 4, 5, 6); - Gestione della memoria: stack e heap. Metodi ricorsivi (lezioni 4 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Eccezioni (lezioni 6 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Implementazione dei tipi generici (lezioni 4 h, obiettivi formativi 2, 3).
Metodi didattici	Le lezioni frontali consistono in 70 ore di lezioni teoriche e 10 ore di esercitazioni in aula. Le lezioni sono dedicate all'illustrazione di: elementi fondamentali di un linguaggio di programmazione; strutture di controllo, astrazioni, compilazione ed esecuzione del linguaggio Java. Il laboratorio consiste in 24 ore, di cui 4 ore sono dedicate all'apprendimento degli strumenti di sviluppo e le restanti allo sviluppo guidato di applicazioni.
Modalità di verifica dell'apprendimento	La modalità di verifica consiste in una prova scritta. Lo scopo di tale prova è quello di verificare l'apprendimento delle conoscenze ed abilità descritte nella Sezione 'Obiettivi del Corso' e di valutare la capacità dello studente di applicare tali elementi alla risoluzione di semplici problemi. La prova scritta (della durata indicativa di 120 minuti) consiste di: Esercizi di natura pratica, il cui scopo è quello di verificare le capacità dello studente di rappresentare i metodi di risoluzione di semplici problemi attraverso l'impiego del linguaggio di programmazione utilizzato. Gli esercizi sono in numero di 6-7. Domande di natura teorica, il cui scopo è di verificare le capacità comunicative sviluppate dallo studente, in particolare l'acquisizione della terminologia tecnica necessaria alla comprensione e comunicazione delle metodologie e dei modelli relativi al linguaggio di programmazione studiato. Le domande sono in numero di 2-3. Il voto della prova scritta è espresso in trentesimi. La prova è superata se il voto è maggiore o uguale a 18/30.
Testi di riferimento	Libro di testo: M. Ferrari e G. Pighizzini. Dai Fondamenti agli oggetti, corso di programmazione Java. Quarta edizione, Pearson Addison-Wesley, 2015. Materiale disponibile sul sito di e-learning d'ateneo: Esercizi con soluzione. Temi d'esame svolti. Esempi di applicazioni.
Altre informazioni	Il docente riceve su appuntamento tramite email a alberto.trombetta@uninsubria.it. Il docente risponde solo a email firmate e provenienti dal dominio studenti.uninsubria.it.

PROGRAMMAZIONE – COMO

Dott.ssa Alessandra Rizzardi

Obiettivi formativi	Questo corso fornisce un'introduzione alla programmazione dei calcolatori mediante il linguaggio Java. L'obiettivo primario è quello di fornire agli studenti le conoscenze necessarie alla comprensione e all'utilizzo del linguaggio Java e all'applicazione del linguaggio alla risoluzione di problemi. Il corso affianca quindi all'insegnamento teorico del linguaggio Java e delle sue strutture, esercitazioni e laboratori. Al termine di questo corso lo studente: 1. Conosce i principali elementi che caratterizzano un linguaggio di programmazione (sintassi, semantica, meccanismi implementativi); 2. Conosce le strutture di controllo, le strutture dati di base e le astrazioni principali
---------------------	---

	fornite dal linguaggio Java e in generale dai linguaggi ad oggetti; 3. Possiede le conoscenze necessarie per affrontare individualmente lo studio di un linguaggio di programmazione e per comprendere i problemi relativi allo sviluppo e all'applicazione dei linguaggi di programmazione. 4. È in grado di applicare il linguaggio di programmazione alla risoluzione di semplici problemi individuando le astrazioni più adatte alla formalizzazione del problema in un linguaggio ad oggetti. 5. È in grado di individuare lo schema concettuale di risoluzione di un problema e di utilizzare il linguaggio di programmazione per l'organizzazione della sua soluzione. 6. È in grado di comunicare in modo compiuto e con la corretta terminologia le caratteristiche di un linguaggio di programmazione e la struttura delle applicazioni realizzate.
Prerequisiti	Non sono richieste conoscenze informatiche/matematiche specifiche oltre quelle fornite da una qualsiasi scuola superiore.
Contenuti	L'acquisizione delle diverse conoscenze ed abilità attese si svilupperà in modo parallelo lungo tutto l'insegnamento, in cui verranno trattati i seguenti argomenti: - Introduzione ai linguaggi di programmazione: astrazioni, algoritmi, sintassi, semantica e strumenti di sviluppo (lezioni 3 h, obiettivo formativo 1); - Introduzione al linguaggio Java. Concetti di base, compilazione, esecuzione, struttura applicazioni (lezioni 3 h, obiettivi formativi 1,2); - Tipi primitivi ed espressioni (lezioni 8 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 1, 2, 3); - Strutture di controllo (lezioni 8 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 1, 2, 3); - Array (lezioni 4 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4); - Gerarchia dei tipi riferimento, cast e polimorfismo (lezioni 6 h, esercitazioni 1 h, laboratorio 4 h, obiettivi formativi 2, 3, 4, 5, 6); - Implementazione delle classi (lezioni 8 h, esercitazioni 1 h, laboratorio ore 4 h, obiettivi formativi 2, 3, 4, 5, 6); - Implementazione dell'ereditarietà (lezioni 8 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Interfacce (lezioni 2 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Variabili, scope e adombramento (lezioni ore 2; esercitazioni 1 h; laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Early- e late-binding in Java (lezioni 4 h esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Gestione della memoria: stack e heap. Metodi ricorsivi (lezioni 4 h, esercitazioni 1 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Eccezioni (lezioni 6 h, laboratorio 2 h, obiettivi formativi 2, 3, 4, 5, 6); - Implementazione dei tipi generici (lezioni 4 h, obiettivi formativi 2, 3).
Metodi didattici	Da definire
Modalità di verifica dell'apprendimento	Da definire
Testi di riferimento	Da definire

Altre informazioni	Da definire
--------------------	-------------

ANALISI MATEMATICA - VARESE

Dott.ssa Andreano Federica

Obiettivi formativi	L'insegnamento ha l'obiettivo di far conseguire allo studente i seguenti risultati di apprendimento: - Con riferimento alla conoscenza e capacità di comprensione: - Saper riconoscere e definire con precisione le principali nozioni incontrate nel corso; - Saper riconoscere e descrivere con precisione i principali risultati e tecniche presentati nel corso; - Saper fornire esempi, controesempi e interpretazioni relativi ai punti precedenti. - Con riferimento alla conoscenza e capacità di comprensione applicate: - Saper calcolare: essere in grado di svolgere le principali operazioni dell'analisi matematica, come risolvere equazioni e disequazioni, svolgere calcoli con i numeri complessi, calcolare limiti, derivate, integrali; - Saper stabilire le proprietà dei principali enti dell'analisi matematica, come funzioni, successioni e serie numeriche, utilizzando le tecniche di calcolo apprese; - Saper applicare: essere in grado di utilizzare le tecniche di calcolo e le proprietà indicate nei punti precedenti per le principali applicazioni geometriche e fisiche, come determinare rette tangenti, risolvere problemi di massimo o minimo, calcolare aree. Nota: gli esempi riportati non sono necessariamente esaustivi.
Prerequisiti	Equazioni e disequazioni. Divisione di polinomi. Trigonometria. Geometria analitica (retta e parabola)
Contenuti	Insiemi e insiemi numerici – Funzioni - Limiti di funzioni – Successioni -Funzioni continue di una variabile reale - Calcolo differenziale – Integrazione - Serie numeriche - Numeri complessi
Metodi didattici	Da definire
Modalità di verifica dell'apprendimento	Da definire
Testi di riferimento	Da definire
Altre informazioni	Da definire

ANALISI MATEMATICA - COMO

Docente non ancora definito

Obiettivi formativi	L'insegnamento ha l'obiettivo di far conseguire allo studente i seguenti risultati di apprendimento: - Con riferimento alla conoscenza e capacità di comprensione: - Saper riconoscere e definire con precisione le principali nozioni incontrate nel corso; - Saper riconoscere e descrivere con precisione i principali risultati e tecniche presentati nel corso; - Saper fornire esempi, controesempi e interpretazioni relativi ai punti precedenti. - Con riferimento alla conoscenza e capacità di comprensione applicate: - Saper calcolare: essere in grado di svolgere le principali operazioni dell'analisi
---------------------	--

	matematica, come risolvere equazioni e disequazioni, svolgere calcoli con i numeri complessi, calcolare limiti, derivate, integrali; b. Saper stabilire le proprietà dei principali enti dell'analisi matematica, come funzioni, successioni e serie numeriche, utilizzando le tecniche di calcolo apprese; c. Saper applicare: essere in grado di utilizzare le tecniche di calcolo e le proprietà indicate nei punti precedenti per le principali applicazioni geometriche e fisiche, come determinare rette tangenti, risolvere problemi di massimo o minimo, calcolare aree. Nota: gli esempi riportati non sono necessariamente esaustivi.
Prerequisiti	Equazioni e disequazioni. Divisione di polinomi. Trigonometria. Geometria analitica (retta e parabola)
Contenuti	Insiemi e insiemi numerici – Funzioni - Limiti di funzioni – Successioni -Funzioni continue di una variabile reale - Calcolo differenziale – Integrazione - Serie numeriche - Numeri complessi
Metodi didattici	Da definire
Modalità di verifica dell'apprendimento	Da definire
Testi di riferimento	Da definire
Altre informazioni	Da definire

ALGORITMI E STRUTTURE DATI – VARESE

Prof. Paolo Massazza

Obiettivi formativi	Il corso ha lo scopo di rendere gli studenti capaci di applicare le conoscenze di base relative alle principali strutture dati e ai principali algoritmi associati. A tale scopo gli studenti apprenderanno le tecniche di base per la progettazione e l'analisi degli algoritmi, insieme alla capacità di risolvere i più classici problemi legati all'elaborazione dei dati. Al termine del corso, lo studente sarà in grado di: 1. Comprendere le caratteristiche rilevanti di un modello di calcolo astratto, nonché l'importanza della complessità di un algoritmo al fine del suo utilizzo concreto. 2. Individuare gli algoritmi e le strutture dati di base più indicate in un dato contesto applicativo. 3. Conoscere e applicare i principali paradigmi di progettazione di algoritmi. Lo studente dovrà inoltre conseguire una consapevole autonomia di giudizio con riferimento alle problematiche tipiche della progettazione di algoritmi efficienti. Lo studente svilupperà infine una proprietà di linguaggio tale da poter formalizzare un problema in modo idoneo a una sua trattazione informatica.
Prerequisiti	È richiesta la capacità di programmare in ambiente "sequenziale"; specificamente è necessario che lo studente padroneggi la programmazione in Java (il linguaggio usato nel corso) nonché gli elementi di base relativi all'architettura di un elaboratore. Le conoscenze e abilità necessarie per un proficuo apprendimento di questo insegnamento sono impartite nei corsi fondamentali del primo anno di Programmazione e di Architetture degli elaboratori.
Contenuti	Le lezioni affronteranno i seguenti argomenti: Modelli di calcolo e complessità (14 h, obiettivo formativo 1) - Formalizzazione dei problemi, complessità degli algoritmi, notazioni asintotiche (6h)

	- Modelli di calcolo (RAM, RASP) e criteri di costo (6h) - Caratteristiche dei linguaggi procedurali (2h) Algoritmi e strutture dati di base (20 h, obiettivo formativo 2) - Strutture dati elementari (vettori, matrici, liste, pile, code) (6h) - Grafi e alberi, rappresentazioni e algoritmi di visita (10h) - Il problema dell'ordinamento: aspetti generali e algoritmi elementari (4h) Algoritmi e strutture dati avanzati (28 h, obiettivo formativo 2) - Ordinamento digitale e algoritmi d'ordinamento avanzati (6h) - Heap e Heapsort (2h) - Tabelle hash (4h) - Alberi binari di ricerca (6h) - Alberi bilanciati (2-3, 2-3-4, red-black) (8h) - Operazioni su partizioni (Union e Find) (2h) Tecniche di progettazione (10 h, obiettivo formativo 3) - Metodologia Divide et Impera (2h) - Programmazione dinamica (4h) - Algoritmi greedy (4h) Gli argomenti verranno affrontati usando come riferimento il linguaggio di programmazione Java. Ciò nondimeno, molti degli argomenti trattati nel corso sono di validità generale, e le tecniche proposte sono applicabili con linguaggi diversi.
Metodi didattici	Il corso si articola in lezioni frontali (72 ore). Ogni lezione presenta sia elementi teorici sia immediate applicazioni ed esempi. Il corso rende disponibile preventivamente tutto il materiale didattico e invita lo studente ad essere presente in aula dopo aver preso visione del materiale della lezione, che verrà svolta in modo tale da aumentare interazione, discussione e di conseguenza apprendimento.
Modalità di verifica dell'apprendimento	L'obiettivo della prova d'esame è l'accertamento dell'acquisizione delle conoscenze e delle abilità descritte nella sezione "Obiettivi formativi", valutando il livello di conoscenza e soprattutto la capacità di mettere in pratica, anche integrandole tra loro, le tecniche di progettazione viste a lezione. L'esame consiste in una prova scritta da svolgersi in aula, a cui fa seguito una prova orale nel caso di esito positivo. La prova scritta–della durata indicativa di 120 minuti–prevede una serie di 5 quesiti relativi agli argomenti trattati a lezione (6 punti disponibili per ogni quesito). L'esito positivo (valutato in trentesimi) della prova scritta permette l'accesso alla successiva prova orale. Tale prova consiste in una visione congiunta della prova scritta in cui l'allievo viene informato sui criteri di correzione e chiamato a fornire eventuali precisazioni, permettendo così al docente di verificare la correttezza della votazione assegnata, apportando nel caso variazioni. La conoscenza della terminologia specifica di dominio viene testata implicitamente, poiché domande e specifiche dei problemi utilizzano tale terminologia. Similmente, l'autonomia di giudizio viene rilevata attraverso quesiti che richiedono scelte ragionate di algoritmi e strutture dati in funzione del contesto.
Testi di riferimento	In aggiunta alle slide e alle dispense distribuite tramite la piattaforma di e-learning, il testo di riferimento è (versione italiana e inglese) • R. Sedgewick, Algoritmi in Java (3ed.), Pearson Education Italia. • R. Sedgewick, K. Wayne, Algorithms, 4th Edition, Addison-Wesley Professional 2011.

	Una trattazione alternativa ed approfondita degli stessi argomenti si può trovare anche su Thomas H. Cormen - Charles E. Leiserson - Ronald L. Rivest - Clifford Stein, Introduzione agli algoritmi e strutture dati, McGraw Hill education, 2010.
Altre informazioni	Il docente riceve su appuntamento, previa richiesta via e-mail a paolo.massazza@uninsubria.it . Il docente risponde solo alle e-mail firmate e provenienti dal dominio studenti.uninsubria.it.

ALGORITMI E STRUTTURE DATI – COMO

Docente non ancora definito

Obiettivi formativi	Il corso ha lo scopo di rendere gli studenti capaci di applicare le conoscenze di base relative alle principali strutture dati e ai principali algoritmi associati. A tale scopo gli studenti apprenderanno le tecniche di base per la progettazione e l'analisi degli algoritmi, insieme alla capacità di risolvere i più classici problemi legati all'elaborazione dei dati. Al termine del corso, lo studente sarà in grado di: 1. Comprendere le caratteristiche rilevanti di un modello di calcolo astratto, nonché l'importanza della complessità di un algoritmo al fine del suo utilizzo concreto. 2. Individuare gli algoritmi e le strutture dati di base più indicate in un dato contesto applicativo. 3. Conoscere e applicare i principali paradigmi di progettazione di algoritmi. Lo studente dovrà inoltre conseguire una consapevole autonomia di giudizio con riferimento alle problematiche tipiche della progettazione di algoritmi efficienti. Lo studente svilupperà infine una proprietà di linguaggio tale da poter formalizzare un problema in modo idoneo a una sua trattazione informatica.
Prerequisiti	È richiesta la capacità di programmare in ambiente "sequenziale"; specificamente è necessario che lo studente padroneggi la programmazione in Java (il linguaggio usato nel corso) nonché gli elementi di base relativi all'architettura di un elaboratore. Le conoscenze e abilità necessarie per un proficuo apprendimento di questo insegnamento sono impartite nei corsi fondamentali del primo anno di Programmazione e di Architetture degli elaboratori.
Contenuti	Le lezioni affronteranno i seguenti argomenti: Modelli di calcolo e complessità (14 h, obiettivo formativo 1) - Formalizzazione dei problemi, complessità degli algoritmi, notazioni asintotiche (6h) - Modelli di calcolo (RAM, RASP) e criteri di costo (6h) - Caratteristiche dei linguaggi procedurali (2h) Algoritmi e strutture dati di base (20 h, obiettivo formativo 2) - Strutture dati elementari (vettori, matrici, liste, pile, code) (6h) - Grafi e alberi, rappresentazioni e algoritmi di visita (10h) - Il problema dell'ordinamento: aspetti generali e algoritmi elementari (4h) Algoritmi e strutture dati avanzati (28 h, obiettivo formativo 2) - Ordinamento digitale e algoritmi d'ordinamento avanzati (6h) - Heap e Heapsort (2h) - Tabelle hash (4h) - Alberi binari di ricerca (6h)

	- Alberi bilanciati (2-3, 2-3-4, red-black) (8h) - Operazioni su partizioni (Union e Find) (2h) Tecniche di progettazione (10 h, obiettivo formativo 3) - Metodologia Divide et Impera (2h) - Programmazione dinamica (4h) - Algoritmi greedy (4h) Gli argomenti verranno affrontati usando come riferimento il linguaggio di programmazione Java. Ciò nondimeno, molti degli argomenti trattati nel corso sono di validità generale, e le tecniche proposte sono applicabili con linguaggi diversi.
Metodi didattici	Da definire
Modalità di verifica dell'apprendimento	Da definire
Testi di riferimento	Da definire
Altre informazioni	Da definire

INGLESE – VARESE

Dott. Daniel Russo

Obiettivi formativi	Il corso si prefigge di sviluppare le competenze ricettive e produttive in lingua inglese per una conoscenza a livello intermedio-superiore (Livello B2 – Upper-intermediate nella griglia di valutazione del Quadro Comune Europeo di Riferimento per le Lingue – QCERL) sia per l'abilità di scrittura che di parlato e ascolto, oltre ad avviare gli studenti all'uso dell'inglese specialistico per l'informatica. In particolare, lo studente dovrà dimostrare di avere acquisito le seguenti competenze e abilità: 1. usare le principali strutture morfo-sintattiche dell'inglese a livello intermedio-superiore nei differenti registri linguistici; 2. riconoscere e usare il lessico specialistico base per l'informatica (hardware, software, programmazione, web, ecc.) per leggere e comprendere i testi specialistici; 3. sostenere una conversazione su temi generali quotidiani o specifici dell'informatica.
Prerequisiti	Livello B1 del Quadro Comune Europeo di Riferimento per le Lingue – QCERL.
Contenuti	Il corso si sviluppa su tre moduli: A) Upper Intermediate language practice (24 ore, obiettivi 1 e 3). In questo modulo vengono sviluppate le competenze lessico-grammaticali a livello intermedio- superiore. Tutte le abilità di base della lingua (lettura, scrittura, parlato e comprensione) verranno pertanto esercitate. In classe ci si concentrerà in particolare su speaking e listening. Verranno affrontati i seguenti argomenti: Time and Tense - The simple present - The present continuous - The past simple - The past continuous - Used to and would - The present perfect simple - The present perfect continuous - The past perfect

	- Going to and will - Present tenses for talking about the future - Modals Sentences - Word order - Reported speech and reporting verbs - Relative clauses Other verb forms - Verbs followed by the infinitive or the –ing form - First, second and zero conditional sentences - The third conditional, wishes and regrets - The passive - Get/have something done - Phrasal verbs Naming and describing - Articles - Personal, possessive, reflexive and demonstrative pronouns - Countable and uncountable nouns and expressions of quantity Functional Areas - Ways of comparing - Possibility in the present and in the past - Present, past and future obligation and necessity B) Linguaggio informatico (16 ore, obiettivi 2 e 3). In questo modulo, mediante l'analisi di testi autentici e la partecipazione ad attività comunicative orali di parlato e comprensione (role play, listening comprehension) e scritte (esercizi funzionali di inglese per scopi speciali, brevi traduzioni), gli studenti vengono introdotti all'uso dell'inglese per l'informatica secondo i seguenti contenuti: - Hardware, input/output devices, storage devices, - Software, graphic software, desktop publishing - Internet, websites, email, web applications - Programming - Gaming C) Presentation (8 ore, obiettivo 3). Gli studenti apprenderanno le strategie linguistiche e comunicative per costruire una presentazione documentaristica su supporto visivo (PowerPoint, Prezi ecc.) relativa a un argomento informatico.
Metodi didattici	Da definire
Modalità di verifica dell'apprendimento	Da definire
Testi di riferimento	Da definire
Altre informazioni	Da definire

INGLESE – COMO

Docente non ancora definito

Obiettivi formativi	Il corso si prefigge di sviluppare le competenze ricettive e produttive in lingua inglese
---------------------	---

	per una conoscenza a livello intermedio-superiore (Livello B2 – Upper-intermediate nella griglia di valutazione del Quadro Comune Europeo di Riferimento per le Lingue – QCERL) sia per l'abilità di scrittura che di parlato e ascolto, oltre ad avviare gli studenti all'uso dell'inglese specialistico per l'informatica. In particolare, lo studente dovrà dimostrare di avere acquisito le seguenti competenze e abilità: 1. usare le principali strutture morfo-sintattiche dell'inglese a livello intermedio-superiore nei differenti registri linguistici; 2. riconoscere e usare il lessico specialistico base per l'informatica (hardware, software, programmazione, web, ecc.) per leggere e comprendere i testi specialistici; 3. sostenere una conversazione su temi generali quotidiani o specifici dell'informatica.
Prerequisiti	Livello B1 del Quadro Comune Europeo di Riferimento per le Lingue – QCERL.
Contenuti	Il corso si sviluppa su tre moduli: A) Upper Intermediate language practice (24 ore, obiettivi 1 e 3). In questo modulo vengono sviluppate le competenze lessico-grammaticali a livello intermedio- superiore. Tutte le abilità di base della lingua (lettura, scrittura, parlato e comprensione) verranno pertanto esercitate. In classe ci si concentrerà in particolare su speaking e listening. Verranno affrontati i seguenti argomenti: Time and Tense - The simple present - The present continuous - The past simple - The past continuous - Used to and would - The present perfect simple - The present perfect continuous - The past perfect - Going to and will - Present tenses for talking about the future - Modals Sentences - Word order - Reported speech and reporting verbs - Relative clauses Other verb forms - Verbs followed by the infinitive or the –ing form - First, second and zero conditional sentences - The third conditional, wishes and regrets - The passive - Get/have something done - Phrasal verbs Naming and describing - Articles - Personal, possessive, reflexive and demonstrative pronouns - Countable and uncountable nouns and expressions of quantity Functional Areas - Ways of comparing

	- Possibility in the present and in the past - Present, past and future obligation and necessity B) Linguaggio informatico (16 ore, obiettivi 2 e 3). In questo modulo, mediante l'analisi di testi autentici e la partecipazione ad attività comunicative orali di parlato e comprensione (role play, listening comprehension) e scritte (esercizi funzionali di inglese per scopi speciali, brevi traduzioni), gli studenti vengono introdotti all'uso dell'inglese per l'informatica secondo i seguenti contenuti: - Hardware, input/output devices, storage devices, - Software, graphic software, desktop publishing - Internet, websites, email, web applications - Programming - Gaming C) Presentation (8 ore, obiettivo 3). Gli studenti apprenderanno le strategie linguistiche e comunicative per costruire una presentazione documentaristica su supporto visivo (PowerPoint, Prezi ecc.) relativa a un argomento informatico.
Metodi didattici	Da definire
Modalità di verifica dell'apprendimento	Da definire
Testi di riferimento	Da definire
Altre informazioni	Da definire

LABORATORIO INTERDISCIPLINARE A E B – *Modulo di Laboratorio interdisciplinare A – VARESE*

Prof. Alberto Coen Porisini

Obiettivi formativi	L'obiettivo principale di questo insegnamento è lo sviluppo di competenze riguardanti la realizzazione di applicazioni software, risultanti dall'applicazione delle conoscenze teoriche acquisite durante il I anno di corso con gli insegnamenti di "Programmazione", "Architetture degli elaboratori" e "Algoritmi e strutture dati". Al termine dell'insegnamento lo studente avrà sperimentato, e sarà pertanto in grado di gestire la progettazione di algoritmi finalizzati alla risoluzione efficiente di problemi complessi, basati su strutture dati adatte a supportare il carico computazionale del problema affrontato
Prerequisiti	L'insegnamento non ha prerequisiti obbligatori, ma per un proficuo svolgimento sono necessarie le conoscenze impartite negli insegnamenti di Programmazione, Architetture degli elaboratori, Algoritmi e strutture dati.
Contenuti	L'acquisizione delle diverse conoscenze ed abilità attese si svilupperà in modo parallelo lungo tutto l'insegnamento. Durante il corso verrà presentato un progetto e verranno discussi i passi principali dello sviluppo di un progetto.
Metodi didattici	La presentazione dei progetti ha lo scopo di esporre le specifiche dei programmi che gli studenti dovranno realizzare. La presentazione avviene in modo interattivo, simulando almeno in parte l'interazione con lo "stakeholder" o utente dell'applicazione. In questo senso, lo studente non è passivamente oggetto della presentazione, ma viene coinvolto per farsi parte attiva ed esercitare le sue conoscenze per individuare ed esplorare i punti salienti e critici dei requisiti.
Modalità di verifica	La prova finale consiste nella realizzazione di un progetto e nella sua discussione. Lo

dell'apprendimento	scopo dell'attività progettuale è quello di verificare la capacità dello studente di comprendere i problemi proposti e, alla luce delle conoscenze acquisite, progettare e realizzare l'applicazione in modo da soddisfare i requisiti. Lo scopo della discussione è quello di verificare la validità delle scelte progettuali ed implementative effettuate e la capacità di comunicare e argomentare tali scelte. Il voto è espresso in trentesimi. La prova finale si intende superata se lo studente ha ottenuto una votazione non inferiore ai 18/30.
Testi di riferimento	Non sono previsti testi o materiali specifici. Gli studenti possono far riferimento ai testi e ai materiali dei corsi di Programmazione, Architetture degli elaboratori, Algoritmi e strutture dati. Se necessario allo svolgimento di uno specifico progetto, viene fornita documentazione ad hoc sul sito dell'e-learning dell'Università.
Altre informazioni	Il docente riceve su appuntamento, previa richiesta via e-mail a alberto.coenporisini@uninsubria.it.

LABORATORIO INTERDISCIPLINARE A E B – *Modulo di Laboratorio interdisciplinare A – COMO*

Prof. Alberto Coen Porisini

Obiettivi formativi	L'obiettivo principale di questo insegnamento è lo sviluppo di competenze riguardanti la realizzazione di applicazioni software, risultanti dall'applicazione delle conoscenze teoriche acquisite durante il I anno di corso con gli insegnamenti di "Programmazione", "Architetture degli elaboratori" e "Algoritmi e strutture dati". Al termine dell'insegnamento lo studente avrà sperimentato, e sarà pertanto in grado di gestire la progettazione di algoritmi finalizzati alla risoluzione efficiente di problemi complessi, basati su strutture dati adatte a supportare il carico computazionale del problema affrontato
Prerequisiti	L'insegnamento non ha prerequisiti obbligatori, ma per un proficuo svolgimento sono necessarie le conoscenze impartite negli insegnamenti di Programmazione, Architetture degli elaboratori, Algoritmi e strutture dati.
Contenuti	L'acquisizione delle diverse conoscenze ed abilità attese si svilupperà in modo parallelo lungo tutto l'insegnamento. Durante il corso verrà presentato un progetto e verranno discussi i passi principali dello sviluppo di un progetto.
Metodi didattici	La presentazione dei progetti ha lo scopo di esporre le specifiche dei programmi che gli studenti dovranno realizzare. La presentazione avviene in modo interattivo, simulando almeno in parte l'interazione con lo "stakeholder" o utente dell'applicazione. In questo senso, lo studente non è passivamente oggetto della presentazione, ma viene coinvolto per farsi parte attiva ed esercitare le sue conoscenze per individuare ed esplorare i punti salienti e critici dei requisiti.
Modalità di verifica dell'apprendimento	La prova finale consiste nella realizzazione di un progetto e nella sua discussione. Lo scopo dell'attività progettuale è quello di verificare la capacità dello studente di comprendere i problemi proposti e, alla luce delle conoscenze acquisite, progettare e realizzare l'applicazione in modo da soddisfare i requisiti. Lo scopo della discussione è quello di verificare la validità delle scelte progettuali ed implementative effettuate e la capacità di comunicare e argomentare tali scelte. Il voto è espresso in trentesimi. La prova finale si intende superata se lo studente ha



	ottenuto una votazione non inferiore ai 18/30.
Testi di riferimento	Non sono previsti testi o materiali specifici. Gli studenti possono far riferimento ai testi e ai materiali dei corsi di Programmazione, Architetture degli elaboratori, Algoritmi e strutture dati. Se necessario allo svolgimento di uno specifico progetto, viene fornita documentazione ad hoc sul sito dell'e-learning dell'Università.
Altre informazioni	Il docente riceve su appuntamento, previa richiesta via e-mail a alberto.coenporisini@uninsubria.it .