



# UNIVERSITÀ DEGLI STUDI DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE

DIPARTIMENTO DI INGEGNERIA ELETTRICA E  
DELLE TECNOLOGIE DELL'INFORMAZIONE

**GUIDA DELLO STUDENTE**

CORSO DI LAUREA IN INFORMATICA

Classe delle Lauree in Scienze e Tecnologie Informatiche L31

## ANNO ACCADEMICO 2020/2021

Vers. 1.0

AGGIORNATA: 22/09/2020

### Sommario

Revisioni.....	3
Introduzione .....	3
Requisiti d'ingresso.....	5
Test di ingresso .....	5
Obiettivi formativi .....	5
Condizione occupazionale .....	8
Regolamento vigente .....	9
Percorso didattico .....	10
Piani di studio .....	11
Manifesto degli Studi del Corso di Laurea in Informatica .....	11
Manifesto degli Studi del Corso di Laurea antecedente l'A.A. 2018-19.....	14
Tirocinio .....	15
Apple Academy .....	17
Esame finale e attribuzione del punteggio finale.....	17
Programma Erasmus .....	18
Calendario Attività Didattiche .....	20
Referenti del Corso di Studi.....	21
Vademecum studenti DIETI .....	21
Schede degli Insegnamenti.....	22

Per ulteriori dettagli si veda il Regolamento (<http://informatica.dieta.unina.it/>).

## Revisioni

Data Aggiornamento	Versione	Descrizione
21/09/2020	1.0	Prima versione pubblicata

## Introduzione

Il corso si propone di formare la figura professionale dell'Informatico, e rappresenta il primo livello di un percorso formativo che prosegue con la laurea magistrale in Informatica. Obiettivi specifici del corso sono quelli di assicurare agli studenti una adeguata padronanza di metodi e contenuti scientifici e tecnologici generali volti a fornire una solida cultura di base nonché l'acquisizione di specifiche conoscenze professionali nell'ambito disciplinare dell'Informatica.

Le competenze teoriche e metodologiche intendono mettere i laureati in grado di aggiornarsi indipendentemente nel corso della propria carriera (formazione permanente) in una disciplina quale l'Informatica, soggetta a continua evoluzione. I contenuti coprono altresì aspetti tecnologici, mantenuti costantemente aggiornati. Vengono insegnate metodologie e tecniche per lo sviluppo di software a livello industriale. Le conoscenze pratiche vengono approfondite e raffinate mediante numerosi corsi di laboratorio (su programmazione, algoritmi, architetture degli elaboratori, sistemi operativi e basi di dati). Di norma, il percorso formativo si conclude con un tirocinio presso una delle oltre 150 aziende con cui il Corso di Studi mantiene contatti.

La laurea in Informatica (classe L-31) dà accesso all'Albo degli Ingegneri dell'Informazione, sez.B (quello accessibile ai laureati triennali).

I tipici sbocchi nel mondo del lavoro comprendono posizioni nei team di progettazione e sviluppo di software, e di gestione dei sistemi informatici, oltre alla libera professione.

Il Corso di Laurea in Informatica ha ottenuto anche nell'anno 2019 il *Bollino GRIN*. Il [Bollino GRIN](#), erogato in collaborazione tra GRIN (Gruppo di Informatica - l'associazione dei professori universitari di informatica) e AICA (Associazione Italiana per l'Informatica ed il Calcolo Automatico), certifica la qualità dei contenuti delle lauree triennali e magistrali di informatica (classi L-31 e LM-18).

La certificazione di qualità dei contenuti si basa su un insieme di criteri che definiscono quanta e quale informatica viene insegnata, quanta matematica di aree rilevanti per l'informatica viene insegnata, e quanti docenti di ruolo di informatica sono presenti.

Maggiori informazioni sulla certificazione GRIN possono essere reperite all'indirizzo:

<http://www.grin-informatica.it/opencms/opencms/grin/didattica/bollino.html>

Come comprovato anche dalle certificazioni, i contenuti del CdS e le sue figure professionali di riferimento risultano in linea con le attuali esigenze del mondo produttivo.

L'affermazione è confermata dalle indagini AlmaLaurea, da cui emerge la forte competitività di questo CdS rispetto ai suoi omologhi di altre sedi in termini di livelli occupazionali, attesa del primo impiego,

percentuale di contratti a tempo indeterminato, livelli di retribuzione, soddisfazione degli studenti e utilità percepita.

Per ulteriori dettagli far riferimento al sito web del corso di studi: <http://informatica.dieti.unina.it/>

## Requisiti d'ingresso.

Le conoscenze richieste per il corso di laurea in Informatica comprendono i principi basilari delle Scienze Matematiche:

1. si richiede che l'allievo possieda le conoscenze di aritmetica, algebra, insiemistica e logica, geometria, calcolo e trigonometria conseguite nel triennio finale della scuola secondaria;
2. si richiedono inoltre le conoscenze elementari della lingua inglese relativamente ai principi della traduzione e comprensione di testi scritti semplici.

Inoltre, sono richieste le seguenti capacità:

- la capacità di interpretare il significato di un testo e di sintetizzarlo o di rielaborarlo in forma scritta e orale;
- l'abilità di comprendere e rispondere a quesiti attenendosi strettamente agli elementi forniti;
- la capacità di individuare i dati di un problema pratico e di utilizzarli per pervenire alla risoluzione nella maniera più rapida;
- la capacità di utilizzare le strutture logiche elementari (ad esempio, il significato di implicazione, equivalenza, negazione di una frase, ecc.) in un discorso scritto e orale.

## Test di ingresso

L'immatricolazione al Corso di studi in Informatica prevede un Test di autovalutazione obbligatorio.

Le modalità di iscrizione al test sono quelle indicate nel sito della Scuola politecnica e delle Scienze di base ( <http://www.scuolapsb.unina.it> ). In particolare, il documento che regola le modalità e il calendario dei test di ammissione si trova alla pagina

<http://www.scuolapsb.unina.it/index.php/studiare-al-napoli/ammissione-ai-corsi#tolc>

(Test di ammissione "on line" (TOLC-I) per Corsi di Studio in Ingegneria e in Scienze **non a numero programmato**).

Si ricorda che il Corso di Studi in Informatica è a numero NON programmato e che dunque il mancato superamento del test non pregiudica l'iscrizione al corso di studio. (In particolare, l'iscrizione al corso di studi può anche essere fatta prima di affrontare il test purché lo si affronti successivamente non oltre i mesi di settembre ed ottobre).

**Il mancato superamento del Test comporta comunque l'attribuzione di un debito formativo (OFA, Obbligo Formativo Aggiuntivo) di 3 CFU che è estinto sostenendo con esito positivo prima di ogni altro esame entro il primo anno di corso un esame a scelta tra Analisi matematica I o Programmazione.**

## Obiettivi formativi

I laureati del corso di laurea devono possedere conoscenze nei vari settori delle scienze e tecnologie dell'informazione sia mirate all'uso e alla gestione consapevole di sistemi informatici, sia mirate alla loro utilizzazione

nella progettazione e sviluppo di sistemi informatici. A tale scopo il laureato dovrà acquisire un'adeguata conoscenza dei settori di base dell'informatica nonché dei lineamenti fondamentali e degli strumenti di supporto della matematica. Pertanto, in accordo con le linee guida delle associazioni nazionali (GRIN) ed internazionali (ACM) del settore, il percorso didattico, prevede:

- l'acquisizione di nozioni di base di fisica e di matematica sia discreta sia del continuo;
- la conoscenza dei principi, dei modelli teorici e delle architetture dei sistemi di elaborazione e delle reti di comunicazione;
- la conoscenza e l'utilizzazione dei sistemi operativi;
- l'acquisizione di elementi di analisi e progettazione degli algoritmi e delle strutture dati;
- l'acquisizione delle moderne metodologie di programmazione nonché la conoscenza dei linguaggi di programmazione rappresentativi dei principali paradigmi di programmazione;
- l'assimilazione dei principi per la progettazione e le dei sistemi per la gestione delle basi di dati e le tecnologie correlate;
- l'acquisizione delle tecniche di progettazione e realizzazione di sistemi informatici.

Il percorso didattico prevede l'acquisizione di conoscenze in settori affini anche a carattere interdisciplinare. Il percorso didattico comprende inoltre:

- un congruo numero di crediti sia riservato a corsi di laboratorio oltre a esercitazioni di laboratorio eventualmente previste in altri corsi;
- lo svolgimento di tirocini formativi presso aziende, enti di ricerca, e strutture della pubblica amministrazione o attività progettuali sostitutive.

Si prevede anche l'incentivazione di soggiorni di studio presso università straniere nel quadro di accordi internazionali.

### **Conoscenza e comprensione**

La formazione di base fornisce al laureato triennale in Informatica la conoscenza e la comprensione dei principi e dei linguaggi di base del metodo scientifico e del settore informatico.

Più specificamente, i risultati del processo di apprendimento permettono di conoscere e comprendere il linguaggio tecnico e scientifico, i modelli, i problemi, le tecniche e gli aspetti tecnologici nei seguenti settori:

- architetture dei sistemi di elaborazione e delle reti di comunicazione;
- sistemi operativi;
- algoritmi e strutture dati;
- metodologie di programmazione e linguaggi di programmazione;
- sistemi per la gestione delle basi di dati;
- ingegneria del software.

Dei settori elencati il laureato è in grado di leggere la documentazione (monografie e manualistica) relativa agli aspetti consolidati dello stato dell'arte, e di mantenersi aggiornato.

I risultati vengono conseguiti negli insegnamenti obbligatori delle discipline matematiche, fisiche e informatiche impartiti per circa 160 CFU complessivi, e mediante la trasmissione personalizzata di conoscenze tramite i corsi a scelta e lo svolgimento dello stage o tirocinio per 15 CFU.

I risultati vengono verificati nelle prove individuali di esame associate agli insegnamenti e nella valutazione dell'attività di stage o tirocinio. I risultati vengono conseguiti principalmente mediante le lezioni frontali e le esercitazioni, nonché mediante le ampie attività di laboratorio previste per i principali settori dell'attività formativa caratterizzante (Programmazione, Sistemi operativi, Algoritmi e strutture dati) e, più in generale, nelle attività di progettazione ed esercitazione di gruppo svolte nell'ambito dei principali insegnamenti caratterizzanti. Ulteriori competenze specifiche vengono acquisite durante l'attività di tirocinio finale. Ai laboratori summenzionati sono dedicati 20 CFU. Alle materie di base sono dedicati 45 CFU. Alle materie caratterizzanti 81 CFU. Ai settori affini 18 CFU.

I risultati vengono verificati nelle prove di esame (scritte e/o orali) associate agli insegnamenti, nelle esercitazioni di laboratorio, nella valutazione degli elaborati eventualmente prodotti per le attività di progettazione ed esercitazione, e nella valutazione del tirocinio e della prova finale connessa.

La ripartizione dei crediti sulle materie caratterizzanti segue le indicazioni della Joint ACM-IEEE Task Force, e da diversi anni ottiene il Bollino GRIN: una certificazione di qualità dei contenuti delle lauree triennali e magistrali in informatica (classi L-31 e LM-18) erogata annualmente dal 2004 in collaborazione tra GRIN (Gruppo di Informatica - l'associazione dei professori universitari di informatica) e AICA (Associazione Italiana per l'Informatica ed il Calcolo Automatico). La certificazione di qualità dei contenuti si basa su un insieme di criteri che definiscono quanta e quale informatica viene insegnata, quanta matematica di aree rilevanti per l'informatica viene insegnata, e quanti docenti di ruolo di informatica sono presenti e finalità del Corso di Laurea

#### **Capacità di applicare conoscenza e comprensione**

I risultati del processo di apprendimento comportano l'acquisizione delle seguenti competenze di carattere sia generale che professionale:

- essere in grado di gestire, amministrare e progettare sistemi informatici anche complessi;
- competenze riguardanti l'acquisizione e la formalizzazione dei requisiti del problema tramite interazione con i committenti;
- avere capacità di affrontare e analizzare problemi e di sviluppare sistemi informatici per la loro soluzione;
- autonomia di giudizio Abilità comunicative Capacità di apprendimento.

## Condizione occupazionale

La condizione occupazionale è rilevata attraverso le indagini di AlmaLaurea che raccoglie attraverso indagini annuali informazioni sul livello occupazionale dei laureati. I dati delle indagini sono consultabili pubblicamente su sito

[www.almalaurea.it](http://www.almalaurea.it)

Di seguito vengono riportati i dati occupazionali delle indagini di AlmaLaurea negli ultimi anni con interviste ai laureati ad un anno dalla laurea. I dati sintetizzati nella tabella sottostante, indicano che il Corso di Studi ha piena efficacia formativa e garantisce un ottimo livello occupazionale. I livelli di occupazione sono sensibilmente più alti della media nazionale relativa ai CdS della stessa classe. Per i laureati dell'indagine 2018 il tasso di occupazione def. Istat è del 62,1% a fronte del 58,5% su scala nazionale.

Il tasso di disoccupazione Istat è al 3,4% a fronte del 4,6% su scala nazionale. La percezione dell'efficacia della laurea nel lavoro svolto nell'indagine per i laureati del 2018 è positiva per il 100% degli intervistati. La retribuzione media (prima retribuzione) è in linea con la media nazionale (1263 mensili e 1289 nella media nazionale).

		Condizione occupazionale e formativa (1)					Efficacia della laurea nel lavoro svolto (riferita ai laureati che dichiarano di dedicarsi al lavoro) (3)	
		Lavora e non è iscritto alla laurea di 2° ciclo	Lavora ed è iscritto alla laurea di 2° ciclo	Non lavora ed è iscritto alla laurea di 2° ciclo	Non lavora, non è iscritto alla laurea di 2° ciclo e non cerca lavoro	Non lavora, non è iscritto alla laurea di 2° ciclo e cerca lavoro	Efficace/ molto efficace	Abbastanza efficace
Indagine 2018	CdS	57,8	4,4	31,1	2,2	4,4	64,3	35,7
	Atenei italiani	44,9	10,8	37,4	3,4	3,4	64,1	29,6
Indagine 2017	CdS	59,0	11,5	24,6	1,6	3,3	75,6	19,5
	Atenei italiani	51,7	9,9	29,6	2,9	5,9	60,4	34,4
Indagine 2016	CdS	55,8	7,7	26,9	1,9	7,7	65,6	25,5
	Atenei italiani	50,7	8,6	28,1	4,7	7,9	59,2	32,1
Indagine 2015	CdS	66,7	8,8	10,5	7	8,8	51,2	46,3
	Atenei italiani	51,2	9,3	25,8	5,6	8	57	33,2
Indagine 2014	CdS	67,8	5,1	18,6	1,7	10,2	65,1	30,2
	Atenei italiani	47,2	10,3	26,4	4,0	12,1	57,4	30,0



<b>Indagine 2013</b>	<b>CdS</b>	66,7	4,2	10,4	4,2	14,6	58,8	35,3
	<b>Atenei italiani</b>	49,4	10,2	25,3	2,9	12,1	51,5	32,7

## Regolamento vigente

A partire dall'anno accademico 2018/19 la laurea triennale ha subito una variazione di regolamento che ne ha aggiornato la struttura e i contenuti. Tutti gli anni della laurea attivati nell'anno accademico corrente sono conformi al regolamento della laurea aggiornata. Ciascuno studente segue un corso di studi conforme al regolamento vigente nell'anno della sua immatricolazione e, dunque gli studenti immatricolati a partire dall'A.A 2018/19 seguono il regolamento attualmente vigente e gli studenti immatricolati precedentemente seguono un piano di studi articolato secondo il vecchio regolamento.

Salvo indicazione contraria le informazioni di seguito riportate fanno riferimento al regolamento della laurea in vigore dall'AA. 2018-19.

I regolamenti del Corso di studi possono essere recuperati nel sito del corso di studi all'indirizzo:

<http://informatica.dieti.unina.it/index.php/it/corsi-di-laurea/regolamenti/laurea-triennale>

## Percorso didattico

### Il percorso didattico:

I anno 📖

7 esami (54 crediti) + colloquio Lingua Inglese (3 crediti)

II anno 📖

7 esami (51 crediti) + attività formative a scelta (12 crediti)

III anno 📖

5 esami di cui 1 a scelta (39 crediti) + attività di tirocinio o equivalente (16 crediti) + prova finale (5 crediti)

Laurea



## Piani di studio

I 12 CFU legati agli esami a libera scelta possono essere conseguiti scegliendo liberamente insegnamenti attivati presso l'Ateneo:

- Se tali CFU sono conseguiti attingendo esclusivamente alla lista di esami attivati per il Corso di Laurea in Informatica (Tabella B del Manifesto degli Studi), lo studente **NON** è tenuto a presentare un piano di studi preventivo.
- Se tali CFU sono conseguiti anche in parte con insegnamenti erogati dall'Ateneo ma non attivati per il Corso di Laurea in Informatica, è necessario presentare un piano di studi al Coordinatore del Corso di Studi **entro il 31 ottobre 2020**.

**In assenza di presentazione del piano di studi, la coerenza degli esami sostenuti dallo studente è valutata dalla Segreteria Didattica al momento della registrazione degli esami stessi. Pertanto, è nella responsabilità dello studente una scelta coerente degli insegnamenti a scelta libera. Scelte non coerenti non permettono la registrazione degli esami connessi eventualmente sostenuti.**

**Lo studente è tenuto a rispettare strettamente le propedeuticità degli esami indicate nel manifesto degli studi. L'esame sostenuto per un insegnamento non potrà essere registrato in mancanza degli insegnamenti ad esso propedeutici.**

Nella composizione dei 12 CFU a libera scelta lo studente può scegliere senza vincoli tra insegnamenti del primo e del secondo semestre (due insegnamenti del primo semestre, oppure due insegnamenti del secondo semestre, oppure un insegnamento del primo ed uno del secondo).

## Manifesto degli Studi del Corso di Laurea in Informatica

In ogni anno accademico viene approvato il *Manifesto degli Studi*, che formalmente struttura l'offerta formativa per gli studenti che si immatricolano in quell'anno accademico. Il Manifesto degli Studi definisce obbligatorietà, propedeuticità, e attivazioni degli insegnamenti facoltativi. In tutta la sua carriera, lo studente deve fare riferimento al Manifesto degli Studi dell'anno di immatricolazione.

Di seguito viene riportato il Manifesto degli Studi approvato per l'a.a. 2020-2021.

**Manifesto degli Studi del Corso di Laurea in Informatica**  
**Classe delle lauree in Scienze e Tecnologie Informatiche – Classe L-31**  
**A.A. 2020-2021**

Insegnamento o attività formativa	Modulo	Semestre	CFU	SSD	Tipologia (*)	Propedeuticità
<b>I anno</b>						
Analisi matematica I		1	9	MAT/05	1	
Algebra		1	9	MAT/02	4	
Programmazione		1	9	INF/01	1	
Inglese		1	3		5	
Laboratorio di programmazione		2	6	INF/01	2	
Architettura degli elaboratori		2	9	INF/01	1	
Geometria		2	6	MAT/03	1	
Fisica generale I		2	6	FIS/01	1	
<b>II anno</b>						
Algoritmi e strutture dati I		1	9	INF/01	2	Analisi matematica I, Programmazione
Object orientation		1	6	INF/01	2	Programmazione
Basi di dati I		1	9	INF/01	2	Programmazione
Sistemi operativi I		2	9	INF/01	2	Architettura degli elaboratori
Laboratorio di algoritmi e strutture dati		2	6	INF/01	2	Programmazione, Laboratorio di programmazione
Elementi di informatica teorica		1	6	INF/01	2	
Linguaggi di Programmazione I		2	6	INF/01	2	Programmazione, Laboratorio di programmazione
Esami a libera scelta (vedi Tab. B)			12		3	
<b>III anno</b>						
Laboratorio di Sistemi Operativi		1	8	INF/01	2	Sistemi operativi I Algebra
Ingegneria del Software	A	1	5	INF/01	2	Object orientation Algebra
Ingegneria del Software	B	1	5	INF/01	2	Object orientation Algebra
Esame a scelta da Tab. A			6	INF/01	2	
Reti di calcolatori I		1	6	INF/01	2	Sistemi operativi I Algebra
Calcolo delle probabilità e statistica		2	9	MAT/06	4	Analisi matematica I Algebra
Tirocinio finale			15		7	
Tirocini ed altre attività di orientamento			1		6	
Prova finale			5		5	

Materie a scelta Tabella A	SSD	CFU	Semestre	Propedeuticità
Linguaggi di programmazione II	INF/01	6	2	Object orientation, Linguaggi di programmazione I, Algebra
Tecnologie web	INF/01	6	1	Object orientation, Linguaggi di programmazione I, Algebra



## Manifesto degli Studi del Corso di Laurea antecedente l’A.A. 2018-19

Di seguito viene riportato l’ultimo *Manifesto degli Studi approvato nell’anno accademico 2017-18 per il corso di studi nella struttura precedente la revisione*. Tale manifesto si applica a coloro che si sono immatricolati in anni accademici antecedenti all’AA 2018-19.

### Manifesto degli Studi del Corso di Laurea in Informatica

#### Classe delle lauree in Scienze e Tecnologie Informatiche – Classe L-31

A.A. 2017-2018

Insegnamento o attività formativa	Modulo	Semestre	CFU	SSD	Tipologia (*)	Propedeuticità
<b>I anno</b>						
Analisi matematica I		1	9	MAT/05	1	
Architettura degli elaboratori I	A	1	6	INF/01	1	
Programmazione I		1	6	INF/01	1	
Inglese		1	3		5	
Programmazione II	Lab.	A	6	INF/01	2	Programmazione I
Programmazione II	A	2	6	INF/01	2	
Architettura degli elaboratori I	B	2	6	INF/01	1	
Algebra		2	6	MAT/02	1	
Geometria		2	6	MAT/03	1	
Fisica generale I		2	6	FIS/01	1	
<b>II anno</b>						
Algoritmi e strutture dati I		1	9	INF/01	2	Programmazione II
Laboratorio di Algoritmi e strutture dati		2	6	INF/01	2	Programmazione II
Basi di Dati e Sistemi Informativi I	A	1	6	INF/01	2	Programmazione I
Sistemi Operativi I		1	9	INF/01	2	Programmazione I, Architettura degli elaboratori I
Basi di Dati e Sistemi Informativi I	Lab.	2	6	INF/01	2	Programmazione I
Elementi di informatica teorica		2	6	INF/01	2	
Linguaggi di Programmazione I		2	6	INF/01	2	Programmazione II
Calcolo numerico		1	6	MAT/08	4	
Reti di calcolatori I		2	6	INF/01	4	Sistemi operativi I
<b>III anno</b>						
Calcolo delle probabilità e statistica matematica		1	6	MAT/06	4	
Laboratorio di Sistemi Operativi		1	6	INF/01	2	Programmazione II
Ingegneria del Software		1	9	INF/01	2	Linguaggi di programmazione I
Esame a scelta da Tab. A			6	INF/01	2	
Esami a libera scelta (vedi Tab. B)			12		3	
Tirocinio finale			15		7	
Tirocini ed altre attività di orientamento			1		6	

Prova finale			5		5	
--------------	--	--	---	--	---	--

**(\*) Legenda delle tipologie delle attività formative ai sensi del DM 270/04**

Attività formativa	1	2	3	4	5	6	7
rif. DM270/04	Art. 10 comma 1, a)	Art. 10 comma 1, b)	Art. 10 comma 5, a)	Art. 10 comma 5, b)	Art. 10 comma 5, c)	Art. 10 comma 5, d)	Art. 10 comma 5, e)

Materie a scelta Tabella A	SSD	CFU	Semestre	Propedeuticità
Linguaggi di programmazione II	INF/01	6	2	Linguaggi di programmazione I
Tecnologie web	INF/01	6	1	

**Esami a libera scelta coerenti con gli obiettivi formativi del Corso di Studi (i cui CFU sono pienamente riconosciuti senza previa delibera della Commissione di Coordinamento Didattico)**

Materie a scelta Tabella B	SSD	CFU	Semestre	Propedeuticità
Algoritmi e strutture dati II	INF/01	6	1	Algoritmi e Strutture Dati I, Laboratorio di Algoritmi e Strutture Dati
Analisi matematica II	MAT/05	6	1	Analisi matematica I
Calcolo parallelo e distribuito mod.A	INF/01	6	1	
Calcolo Scientifico mod.A	MAT/08	6	1	Calcolo numerico
Economia ed organizzazione aziendale	ING-IND/35	6	2	
Fisica Generale II	FIS/01	6	2	Fisica Generale I Analisi Matematica I
Griglie computazionali	INF/01	6	2	
Interazione uomo macchina	INF/01	6	2	
Linguaggi di programmazione II	INF/01	6	2	Linguaggi di programmazione I
Logica	M-FIL/02	6	1	
Reti di calcolatori II	INF/01	6	2	Reti di calcolatori I
Ricerca operativa	MAT/09	6	1	Algoritmi e strutture dati I
Sistemi informativi multimediali	INF/01	6	2	
Sistemi Operativi II	INF/01	6	2	Sistemi Operativi I
Tecnologie web	INF/01	6	1	

## Tirocinio

Ai fini del conseguimento della Laurea triennale gli studenti dovranno svolgere un tirocinio, o attività equivalente, connesso al completamento della loro preparazione scientifica e professionale.

È fortemente consigliato che il tirocinio venga svolto presso aziende esterne già convenzionate, o previa stipula di convenzione, con l'Università degli Studi di Napoli "Federico II". Il tirocinio esterno si configura, infatti, a pieno titolo come "esperienza lavorativa pregressa in azienda" per fini curriculari. Gli studenti possono svolgere il tirocinio anche presso istituzioni di ricerca come CNR, INFN, INFN, Osservatorio Astronomico ed altri, ovvero scegliere di lavorare su un progetto proposto da un gruppo di ricerca o da un docente universitario, purché l'addestramento conseguente sia rivolto al fine di una maturazione scientifica e professionale.

In ogni caso, il lavoro richiesto per portare a termine il tirocinio è determinato dal numero di crediti corrispondente, e, di norma, non inferiore a 4 mesi, e si conclude con la stesura di un elaborato obbligatorio che documenti il lavoro svolto. Informazioni dettagliate sui tirocini e sull'iter per l'assegnazione di un tirocinio sono recuperabili su sito del Corso di Laurea all'indirizzo:

<http://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/tirocini/iter-burocratico>

**Tutor accademico.** L'assegnazione di un tirocinio comprende la designazione di un tutor accademico scelto tra i docenti del Corso di Laurea incaricato di assistere lo studente nell'esperienza di tirocinio. Il tutor accademico assiste inoltre lo studente nell'organizzazione e nella stesura dell'elaborato finale che ha come oggetto di norma la presentazione del lavoro svolto dallo studente nel corso della sua esperienza di tirocinio. L'elaborato finale sarà presentato e discusso nel corso dell'esame finale di Laurea.

### **Aziende convenzionate**

Condizione necessaria affinché una azienda possa essere selezionata come sede del tirocinio è l'esistenza di una convenzione specifica (non scaduta) tra l'azienda e l'Università degli Studi di Napoli "Federico II".

La lista delle aziende con le quali intercorrono rapporti di tirocinio è fornita alla pagina

<http://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/tirocini/lista-aziende>.

Nell'indirizzo <http://informatica.dieti.unina.it/index.php/dettaglio-ricerca-tirocini> viene fornito un elenco delle aziende che hanno offerto degli argomenti di tirocinio specifici per laureandi in Informatica. L'elenco comprende, per ogni azienda, una descrizione sintetica di tali offerte. Gli studenti possono utilizzare gli elenchi sopra indicati per avere una indicazione degli argomenti trattati dalle singole aziende.

Qualora lo studente abbia la possibilità di svolgere il tirocinio presso una azienda NON ancora convenzionata con l'Ateneo Federiciano, è necessario, **prima** dell'inizio del tirocinio, richiedere la stipula dell'apposita convenzione.

### **Requisiti richiesti per l'assegnazione del tirocinio**

**Possono richiedere l'assegnazione del tirocinio gli studenti a cui manchino esclusivamente esami del terzo anno.**

In casi di particolare e motivata urgenza, gli studenti possono richiedere una deroga alla commissione tirocini.

### **Tirocinio interno.**

Una alternativa al tirocinio aziendale è il tirocinio interno svolto sotto la tutela di un docente del Corso di Laurea. Le modalità di assegnazione del tirocinio interno sono descritte alla pagina

<http://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/tirocini/iter-burocratico>

Tenuto conto della possibilità di esperienza professionale e della possibilità di un futuro sbocco lavorativo, lo studente è invitato a preferire un tirocinio aziendale rispetto ad un tirocinio interno.



## Apple Accademy

Il Corso di studio riconosce crediti per il Diploma ottenuto presso le iOS Developer Accademy (partnership Università degli studi Federico II e Apple). Tenuto conto delle attività svolte il Corso di Studi ha deliberato il riconoscimento di 15 CFU da riconoscere nell'ambito dell'attività obbligatoria di tirocinio. Lo studente che intende avvalersi dell'esperienza iOS Developer Accademy per il riconoscimento dei crediti di tirocinio dovrà farne richiesta al Coordinatore del Corso di Studio.

## Esame finale e attribuzione del punteggio finale

L'esame finale consiste nella presentazione alla commissione del contenuto dell'elaborato di tesi scritto a partire dall'esperienza del tirocinio (interno o esterno). La presentazione ha una durata massima di 10 minuti.

Al termine dell'esame viene attribuito un punteggio complessivo ottenuto dalla somma dei seguenti addendi:

- a. la media della carriera espressa in centodecimi (sia essa  $X$ );
- b. un incremento della carriera così calcolato: 0 se  $X \leq 77$ , altrimenti  $X \cdot 3 / 22 - 11$ ;
- c. un punteggio compreso tra 1 e 6 assegnato all'elaborato e alla presentazione;
- d. un bonus di incentivazione legato alla velocità della carriera: 5 punti per il conseguimento in 3 anni; 2 punti per il conseguimento in 2 anni.

Il punteggio finale si ottiene come somma dei quattro addendi arrotondata per difetto se la parte decimale è inferiore a 0,5, per eccesso se uguale o superiore a 0,5.

Qualora il voto base derivante dalla carriera del candidato espresso in trentesimi sia uguale o superiore a 28 (ventotto) e il punteggio finale risulti uguale a 110 o superiore, la commissione può considerare l'attribuzione della lode.

## Programma Erasmus

Erasmus+ è il programma di mobilità voluto e finanziato dall'Unione Europea che consente anche agli studenti universitari di trascorrere un periodo di studio presso una Università straniera convenzionata, con un contributo finanziario UE e con la possibilità di seguire corsi, sostenere esami e di usufruire delle strutture disponibili senza pagare a questa le tasse di iscrizione.

L'attività didattica (esami, tirocini) svolta in Erasmus+ viene poi riconosciuta, sia in termini di crediti che di voti, dall'Università di appartenenza.

In formazioni più dettagliate possono essere recuperate nel sito del Corso di Studi all'indirizzo

<http://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/ersmus/informazioni-generalis>

Il referente del Programma Erasmus del Corso di Studi è il Prof. Sergio Di Martino.

**Scambi Erasmus Attivi per l'a.a. 2020/21, Corso di Laurea Triennale in Informatica (N86)****Scambi Erasmus Attivi per l'a.a. 2020/21, Corso di Laurea Triennale in Informatica (N86)**

<b>Nazione</b>	<b>Nome Organismo di ricerca</b>	<b>Codice Erasmus</b>	<b>Titolo Rilasciato</b>
<b>Austria</b>	Technische Universitaet Wien	A WIEN02	Solo Italiano
<b>Francia</b>	Université Francois Rabelais - Tours	F TOURS01	Solo Italiano
<b>Francia</b>	Institut national des Sciences Appliquées - INSA - Centre Val de Loire - Campus Bourges	F BLOIS13	Solo Italiano
<b>Germania</b>	Freie Universitaet Berlin	D BERLIN01	Solo Italiano
<b>Germania</b>	Leibniz Universität Hannover	D HANNOVE01	Solo Italiano
<b>Irlanda</b>	University College Dublin	IRL DUBLIN02	Solo Italiano
<b>Olanda</b>	Rijksuniversiteit Groningen	NL GRONING	Solo Italiano
<b>Polonia</b>	Uniwersytet Jagiellonski (Cracovia)	PL KRAKOW01	Solo Italiano
<b>Repubblica di Macedonia</b>	University for Inf. Sc. and Tech. St. Paul the Aposole Ohrid	MK OHRID01	Solo Italiano
<b>Spagna</b>	Universidad de Granada Es.Tec.Sup. de Ing. Inf. y de Tel.	E GRANADA01	Solo Italiano
<b>Spagna</b>	Universitat Politecnica de Catalunya - FIB	E BARCELO03	Solo Italiano
<b>Spagna</b>	Universitat Rovira i Virgili (Tarragona)	E TARRAGO01	Solo Italiano
<b>Spagna</b>	Universidad de Oviedo	E OVIEDO01	Solo Italiano
<b>Spagna</b>	Universidad de Huelva	E HUELVA01	Solo Italiano
<b>Svezia</b>	Linnaeus University	S VAXJO03	Solo Italiano
<b>Turchia</b>	Yildiz Teknik Universitesi	TR ISTANBU07	Solo Italiano

## Calendario Attività Didattiche

### Calendario delle attività didattiche - a.a. 2020/2021

	Inizio	Termine
<b>1° periodo didattico</b>	28 settembre 2020	22 dicembre 2020
<b>1° periodo di esami</b>	23 dicembre 2020	27 febbraio 2021
<b>Finestra esami marzo</b>	1 marzo 2021	31 marzo 2021
<b>2° periodo didattico</b>	08 marzo 2021	11 giugno 2021
<b>2° periodo di esami</b>	12 giugno 2021	31 luglio 2021
<b>3° periodo di esami</b>	31 agosto 2021	30 settembre 2020
<b>Finestra esami ottobre</b>	1 ottobre 2021	31 ottobre 2021

Appelli garantiti per periodo di esami:

- Due appelli nel 1° periodo di esami ed uno nella Finestra di Marzo;
- Due appelli nel 2° periodo di esami;
- Un appello nel 3° periodo di esami ed uno nella Finestra di Ottobre.

### Orario degli insegnamenti

L'orario degli insegnamenti è consultabile on-line all'indirizzo:

<http://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/orario-lezioni/laurea-triennale>

### Calendario degli Esami di profitto

Il calendario degli esami di profitto è consultabile on-line all'indirizzo:

<http://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/calendario-esami/triennale>

### Calendario degli Esami di Laurea

Il calendario degli esami di laurea è consultabile on-line all'indirizzo:

<http://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/esami-di-laurea/triennale>

## Referenti del Corso di Studi

### **Coordinatore Didattico dei Corsi di Studio in Informatica:**

Prof. Adriano Peron – Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - tel. 081/679280 - e-mail: [adrperon@unina.it](mailto:adrperon@unina.it).

### **Referenti per le attività di tirocinio:**

Prof. Massimo Benerecetti – Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - tel. 081/679XXX - e-mail: [massimo.bennerecetti@unina.it](mailto:massimo.bennerecetti@unina.it).

Prof.ssa Anna Corazza – Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - tel. 081/679XXX - e-mail: [anna.corazza@unina.it](mailto:anna.corazza@unina.it).

Prof. Francesco Cutugno – Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - tel. 081/679XXX - e-mail: [francesco.cutugno@unina.it](mailto:francesco.cutugno@unina.it).

Dott. Fabio Mogavero - Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - tel. 081/679XXX - e-mail: [fabio.mogavero@unina.it](mailto:fabio.mogavero@unina.it).

### **Referente del Corso di Laurea per il Programma ERASMUS+:**

Prof. Sergio Di Martino – Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - tel. 081/679272 - e-mail: [sergio.dimartino@unina.it](mailto:sergio.dimartino@unina.it).

### **Referente sito web del Corso di Studi** (<http://informatica.diети.unina.it/>)

Prof. Luigi Sauro – Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - e-mail: [luigi.sauro@unina.it](mailto:luigi.sauro@unina.it).

### **Referente per l'orientamento del Corso di Studi:**

Prof. Marco Faella – Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - e-mail: [marco.faella@unina.it](mailto:marco.faella@unina.it).

### **Rappresentanti degli studenti:**

Sig. Luigi Coppola – Dipartimento di Ingegneria Elettrica e delle Tecnologie dell'Informazione - e-mail: [luigi.coppola18@studenti.unina.it](mailto:luigi.coppola18@studenti.unina.it).

## Vademecum studenti DIETI

Il Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione ha redatto un Vademecum per gli studenti per orientarsi nelle strutture e nei servizi erogati. Il documento è reperibile al seguente indirizzo:

[http://www.diети.unina.it/index.php?option=com\\_content&view=article&id=849&Itemid=426&lang=it](http://www.diети.unina.it/index.php?option=com_content&view=article&id=849&Itemid=426&lang=it)

## Schede degli Insegnamenti

Le schede di seguito riportate si riferiscono agli insegnamenti previsti dal regolamento aggiornato.

### SCHEDA DELL' INSEGNAMENTO DI *Algebra*

#### TITOLO INSEGNAMENTO IN INGLESE *Algebra*

**Docenti:** Gruppo 1 Antonella Leone, Gruppo 2 Giovanni Cutolo

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
MAT/02	9	X			X		X

**Insegnamenti propedeutici previsti:** nessuno

#### OBIETTIVI FORMATIVI

Obiettivo del corso è quello di fornire allo studente la capacità di utilizzare correttamente il linguaggio insiemistico, migliorare la sua capacità di astrazione e quella di riconoscere strutture matematiche, focalizzando l'attenzione sulle principali strutture algebriche e su quelle della matematica discreta che hanno applicazioni in informatica. Lo studente acquisirà, in particolare, familiarità con l'aritmetica modulare, con le proprietà dei polinomi su campi di ordine primo, e saprà riconoscere e descrivere in dettaglio relazioni di equivalenza, ordinamenti e reticoli, strutture booleane incluse.

#### CONTENUTI

Logica intuitiva, introduzione elementare al calcolo dei predicati. Linguaggio della teoria degli insiemi, applicazioni e confronto tra insiemi. Calcolo combinatorio, fattoriali, coefficienti binomiali. Relazioni binarie: equivalenze e partizioni; ordinamenti, buon ordinamento dei numeri naturali e principio di induzione; introduzione a grafi e alberi. Operazioni e strutture algebriche. Semigrupperi, monoidi, gruppi, anelli, campi. Parti stabili, sottostrutture. Omomorfismi e strutture quoziente. Reticoli, algebre di Boole, anelli booleani. Reticoli come particolari insiemi ordinati e come strutture algebriche. Sottoreticoli, isomorfismi. Connessioni tra reticoli, algebre e anelli di Boole. Aritmetica. L'anello  $\mathbb{Z}$  degli interi, il teorema fondamentale dell'aritmetica, l'algoritmo euclideo delle divisioni successive. Congruenze in  $\mathbb{Z}$ , gli anelli delle classi di resto, aritmetica modulare. Equazioni congruenziali lineari. Polinomi. L'anello dei polinomi a una indeterminata, divisione tra polinomi. Applicazioni polinomiali, radici di un polinomio, teorema di Ruffini e sue conseguenze. Polinomi irriducibili; fattorizzazione di polinomi a coefficienti in un campo.

#### MODALITA' DIDATTICHE

Lezioni frontali. Esercitazioni.

#### MODALITA' DI ESAME

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	

Altro	
-------	--

**SCHEDA DELL' INSEGNAMENTO DI *Algoritmi e Strutture Dati I***

**TITOLO INSEGNAMENTO IN INGLESE *Algorithms and Data Structures I***

**Docenti: Massimo Benerecetti**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	9		X		X		X

**Insegnamenti propedeutici previsti: Analisi matematica I, Programmazione**

**OBIETTIVI FORMATIVI**

Il corso si propone di fornire le conoscenze di base per la progettazione e l'analisi di algoritmi e strutture dati efficienti. In particolare, verranno illustrate le tecniche di base per l'analisi della complessità degli algoritmi e per la valutazione dell'efficienza delle principali strutture dati. Tali concetti verranno illustrati a livello teorico e metodologico e applicati, a titolo esemplificativo, all'analisi di algoritmi specifici per risolvere problemi fondamentali (ad esempio, algoritmi di ordinamenti e di ricerca), di strutture dati elementari (tra cui liste, alberi, grafi) e strutture dati avanzate (come tabelle hash e alberi bilanciati).

**CONTENUTI**

Cenni al calcolo della complessità computazionale degli algoritmi: notazione asintotica; calcolo del tempo di esecuzione di algoritmi iterativi; calcolo del tempo di esecuzione di algoritmi ricorsivi, metodi di soluzione di equazioni di ricorrenza. Analisi di complessità dei principali algoritmi di ordinamento: insertion sort, selection sort, merge sort, heap sort, quick sort. Strutture dati elementari e algoritmi fondamentali: heap, code a priorità, stack, liste puntate, alberi. Alberi binari di ricerca, alberi bilanciati: algoritmi di ricerca, inserimento, cancellazione in alberi binari di ricerca, alberi AVL e alberi Rossi e Neri. Tabelle ad Accesso Diretto e Tabelle Hash. Rappresentazione di grafi e grafi pesati, algoritmi di attraversamento di grafi: algoritmi di visita in ampiezza (BFS) e in profondità (DFS). Applicazioni delle visite di grafi: cammini minimi in grafi non pesati, verifica dell'aciclicità di un grafo orientato, ordinamenti topologici di grafi aciclici, componenti fortemente connesse. Problemi su grafi pesati: albero minimo di copertura, cammini minimi su grafi pesati.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
Altro						



**SCHEDA DELL' INSEGNAMENTO DI *Algoritmi e Strutture Dati II***

**TITOLO INSEGNAMENTO IN INGLESE *Algorithms and Data Structures II***

**Docenti: Massimo Benerecetti**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X			X	

**Insegnamenti propedeutici previsti: Algoritmi e strutture dati I, Laboratorio di Algoritmi e Strutture Dati**

**OBIETTIVI FORMATIVI**

Il corso intende fornire un'introduzione alle tecniche avanzate di progettazione degli algoritmi, alla complessità computazionale e alla trattabilità dei problemi. Vengono, in particolare, presentate le principali tecniche di dimostrazione di correttezza, esaminate le tecniche di progettazione greedy e di programmazione dinamica, con applicazioni alla soluzione di vari problemi di ottimizzazione, di compressione dei dati e problemi su grafi pesati. Vengono introdotte le classi di complessità P e NP e il concetto di NP-completezza e di riduzione tra problemi. Vengono infine presentate tecniche di progettazione ed analisi di algoritmi approssimati e di algoritmi randomizzati.

**CONTENUTI**

Il problema della correttezza degli algoritmi: dimostrazioni per induzione, dimostrazioni di correttezza di algoritmi ricorsivi.

Tecniche di progettazione di algoritmi: introduzione agli algoritmi greedy ed alla programmazione dinamica per la soluzione di problemi di ottimizzazione (ad es., problema dello zaino intero e frazionario, percorsi minimi su grafi pesati, i codici di Huffman, problemi di scheduling). Introduzione alla Teoria della Complessità: problemi trattabili e non trattabili, le principali classi di complessità (P e NP), il concetto di riduzione polinomiale tra problemi e il concetto di NP-completezza, esempi di problemi NP-completi e dimostrazioni di NP-completezza. Introduzione all'intrattabilità computazionale. Introduzione agli algoritmi approssimati; fattore di approssimazione; esempi di algoritmi approssimati per problemi su grafi. Introduzione agli algoritmi randomizzati. Progettazione ed analisi di algoritmi randomizzati per problemi di scheduling e problemi su grafi.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	
Alto						

**SCHEDA DELL' INSEGNAMENTO DI *Analisi Matematica I***

**TITOLO INSEGNAMENTO IN INGLESE *Calculus I***

**Docenti: Gruppo 1 Francesco Oliva, Gruppo 2 Daniele Castorina**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
<b>MAT/05</b>	<b>9</b>	<b>X</b>			<b>X</b>		<b>X</b>

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Pieno possesso della simbologia insiemistica. Consapevolezza della necessità dei vari ampliamenti numerici e delle relative procedure. Conoscenza delle proprietà e dei grafici delle funzioni: lineare, valore assoluto, potenza, esponenziale, logaritmo, trigonometriche e trigonometriche inverse. Conoscenza della definizione di limite, di continuità e del significato geometrico e fisico di derivata. Calcolo di derivate di funzioni. Conoscenza e uso dei Teoremi di Rolle, Lagrange e Cauchy. Calcolo dei limiti con e senza regola di l'Hopital. Studio del grafico di una funzione reale di una variabile reale. Approssimazione di funzioni regolari mediante polinomi. Zeri di una funzione, conoscenza di vari algoritmi. Conoscenza del concetto di serie. Conoscenza del significato geometrico degli integrali definiti. Calcolo di integrali indefiniti.

**CONTENUTI**

Cenni di Teoria degli insiemi. Insiemi numerici: i numeri naturali; i numeri interi; il principio di induzione; i numeri razionali; i numeri reali; funzioni reali di una variabile reale e loro rappresentazione cartesiana; funzioni invertibili e funzione monotone; le funzioni elementari. Estremi inferiore e superiore di insiemi e funzioni. Successioni e loro limiti. Limiti di funzione e funzioni continue. Funzioni continue in un intervallo. Derivate. Massimi e minimi. Criteri di monotonia. Funzioni convesse e concave. Formula di Taylor ed applicazioni. Metodo di Newton. Integrale di Riemann: definizione e proprietà principali. Integrabilità delle funzioni continue. Integrali indefiniti. Teorema fondamentale del calcolo integrale. Formula fondamentale del calcolo integrale. Serie numeriche.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
------------------------------	-----------------	---	--------------	--	------------	--

In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
---	---------------------	--	-------------------	---	-------------------	---

<b>Altro</b>	
--------------	--

**SCHEDA DELL' INSEGNAMENTO DI *Architettura degli Elaboratori***

**TITOLO INSEGNAMENTO IN INGLESE *Computer Architecture***

**Docenti: Gruppo 1 Silvia Rossi, Gruppo 2 Luigi Sauro**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	9	X				X	X

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Conoscere e applicare le principali codifiche digitali dei dati. Saper interpretare e manipolare espressioni dell'algebra di Boole. Saper tradurre un'espressione booleana in circuito combinatorio e viceversa. Saper minimizzare espressioni booleane. Conoscere le macchine di Moore e Mealy. Conoscere la struttura dei principali circuiti logico-aritmetici e delle ALU.

Conoscere l'architettura dei microprocessori basati sul paradigma ARM. Saper realizzare programmi in linguaggio assembly di un processore ARM. Conoscere le principali architetture di memoria, incluse le memorie cache e la memoria virtuale.

**CONTENUTI**

Rappresentazioni digitale dei dati. Operazioni aritmetiche e overflow. Algebra di Boole, funzioni booleane, circuiti combinatori e porte logiche. Minimizzazione di funzioni booleane. Multiplexer e decoder. Elementi di timing. Circuiti sequenziali elementari: latch e flip-flop. Macchine di Mealy e Moore: analisi e sintesi. Circuiti addizionatori e ALU. Architettura ARM: elementi hardware, formato istruzione, architettura interna. Programmazione in assembly ARM. Connessioni con i costrutti del linguaggio C. Introduzione alle memorie cache. Analisi delle prestazioni di sistemi con cache. Introduzione al concetto di memoria virtuale. Traduzione degli indirizzi. Architetture a ciclo singolo, a ciclo multiplo e basate su pipeline.

**MODALITA' DIDATTICHE**

Lezioni frontali ed esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla	X	A risposta libera	X	Esercizi numerici	
Altro						

**SCHEDA DELL' INSEGNAMENTO DI *Basi di dati I***

**TITOLO INSEGNAMENTO IN INGLESE *Data Bases I***

**Docenti: Adriano Peron**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	9		X		X		X

**Insegnamenti propedeutici previsti: Programmazione**

**OBIETTIVI FORMATIVI**

Obiettivo del corso è l'acquisizione delle metodologie per la progettazione e l'implementazione di una base di dati e la predisposizione della sua interfaccia con utenti e/o programmi applicativi. In particolare, lo studente acquisirà le metodologie per strutturare e documentare il progetto; acquisirà gli elementi per la comprensione della struttura, delle funzionalità e degli aspetti tecnologici dei sistemi per la gestione di basi di dati (DBMS) con particolare riferimento a quelli che adottano un modello relazionale dei dati. Lo studente acquisirà conoscenza dei linguaggi standard di interrogazione e manipolazione dei dati per il modello relazionale dei dati.

**CONTENUTI**

Architettura dei Sistemi per la gestione di basi di dati.

Progettazione concettuale di un database, modello dei dati (descritto mediante Class Diagram UML) e vincoli.

Documentazione della progettazione concettuale.

Il modello relazionale dei dati e progettazione logica dei dati.

Algebra relazionale per l'interrogazione delle basi di dati relazionali.

Lo standard SQL99 per:

- la definizione dei dati,
- la definizione dei vincoli di dominio, di ennupla, intra-relazionali e inter-relazionali;
- la interrogazione dei dati e la definizione delle viste;
- l'aggiornamento dei dati;
- l'aggiornamento dei metadati;
- La definizione degli indici.

Introduzione alle tecniche di programmazione per basi di dati:

- Approccio basato su linguaggio di programmazione per basi di dati;
- Approccio basato su chiamate di funzione SQL per la programmazione Java (JDBC);
- I trigger;
- Le procedure memorizzate;
- SQL dinamico.

Aspetti di sicurezza nelle basi di dati: controllo d'accesso, risorse e privilegi.

Cenni sulla tecnologia di un database server: la gestione delle transazioni.

La normalizzazione delle basi di dati relazionali.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	<input checked="" type="checkbox"/>	Solo scritta	<input type="checkbox"/>	Solo orale	<input type="checkbox"/>
------------------------------	-----------------	-------------------------------------	--------------	--------------------------	------------	--------------------------

In caso di prova scritta i quesiti sono	A risposta multipla	<input type="checkbox"/>	A risposta libera	<input checked="" type="checkbox"/>	Esercizi numerici	<input checked="" type="checkbox"/>
---	---------------------	--------------------------	-------------------	-------------------------------------	-------------------	-------------------------------------

Altro						
-------	--	--	--	--	--	--

**SCHEDA DELL' INSEGNAMENTO DI *Calcolo delle Probabilità e Statistica***

**TITOLO INSEGNAMENTO IN INGLESE *Probability Theory and Statistics***

**Docenti: *Luigia Caputo***

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
<b>MAT/06</b>	<b>9</b>			<b>X</b>		<b>X</b>	<b>X</b>

**Insegnamenti propedeutici previsti: Algebra, Analisi matematica I**

**OBIETTIVI FORMATIVI**

L'obiettivo specifico di apprendimento dell'insegnamento è quello dell'acquisizione dei principi e metodi di base del calcolo delle probabilità e della statistica sia descrittiva che inferenziale. L'approccio didattico che si intende utilizzare è prevalentemente quello euristico, in quanto il collocare i contenuti in semplici contesti applicativi può facilitare gli studenti nell'apprendimento del pensiero probabilistico, senza tuttavia tralasciare le dimostrazioni di alcuni risultati teorici alla base della disciplina. Un ulteriore obiettivo è quello di fornire un'iniziale indicazione di come tali risultati teorici del Calcolo delle Probabilità trovino naturale e piena applicazione nella costruzione dei metodi statistici.

**CONTENUTI**

Il problema del contare. Le varie definizioni di probabilità di un evento. Probabilità combinatorie. Esperimento casuale e spazio campione ad esso associato. Il concetto di indipendenza e probabilità condizionata. Il teorema di Bayes. Il concetto di variabile aleatoria, funzione di distribuzione e classificazione. Alcuni modelli notevoli di variabili aleatorie. Funzione di variabile aleatoria e generatori di numeri aleatori. Estensione al caso a più dimensioni. Momenti di una variabile aleatoria. Disuguaglianze notevoli. Convergenze e teoremi fondamentali. Il concetto di campione casuale semplice. Statistiche; statistiche campionarie; statistiche d'ordine. Campioni da generatrici normali. Stima puntuale. Stimatori e relative proprietà. Metodi di costruzione degli stimatori. Cenni alla stima intervallare e ai test d'ipotesi.

**MODALITA' DIDATTICHE**

Lezioni frontali. Allo scopo di verificare l'effettiva acquisizione delle conoscenze trasmesse, sono stimolati interventi, riflessioni e richieste di chiarimenti da parte degli studenti.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	<input type="checkbox"/>	Solo scritta	<input type="checkbox"/>	Solo orale	<input checked="" type="checkbox"/>
In caso di prova scritta i quesiti sono	A risposta multipla	<input type="checkbox"/>	A risposta libera	<input type="checkbox"/>	Esercizi numerici	<input type="checkbox"/>
Altro	Nella prima parte del colloquio sarà richiesta la risoluzione di un quesito a carattere applicativo; lo studente, in maniera autonoma o opportunamente guidato, dovrà dimostrare di sapere inquadrare la					

	<p>questione tra gli argomenti del programma, di saper scegliere le opportune tecniche risolutive e di essere in grado di interpretare correttamente i risultati ottenuti. Ulteriori richieste di carattere teorico tenderanno ad accertare, oltre alla conoscenza dei contenuti presenti nel programma, la consapevolezza dell'impostazione assiomatica nonché il raggiungimento di una sufficiente padronanza del relativo linguaggio.</p>
--	--



**SCHEDA DELL' INSEGNAMENTO DI *Calcolo Numerico***

**TITOLO INSEGNAMENTO IN INGLESE *Numerical Analysis***

**Docenti: Luisa D'Amore**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
<b>MAT/08</b>	<b>6</b>		<b>X</b>		<b>X</b>		<b>X</b>

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Il corso rappresenta una introduzione ai concetti fondamentali della matematica numerica per la risoluzione di problemi matematici che sono modelli di situazioni reali (calcolo scientifico) e si pone, pertanto, i seguenti obiettivi: analisi dei principali metodi che sono alla base della risoluzione numerica di alcune classi di problemi con particolare riguardo alla stabilità e all'efficienza; progettazione di algoritmi risolutivi efficienti ed accurati; sviluppo di tecniche implementative, analisi degli errori e testing.

**CONTENUTI**

Approccio computazionale alla risoluzione di un problema. Sorgenti di errore. Analisi degli errori: Forward e backward. L'aritmetica standard IEEE. Stabilità di un algoritmo numerico. Condizionamento di un problema matematico. Indice di condizionamento. Calcolo matriciale: metodi diretti per matrici piene e strutturate. Metodo di eliminazione di Gauss. Algoritmo di fattorizzazione LU. Stabilità dell'algoritmo di eliminazione di Gauss, strategie di pivoting. Attività di laboratorio.

**MODALITA' DIDATTICHE**

Lezioni frontali ed attività di laboratorio.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	<input checked="" type="checkbox"/>	Solo scritta	<input type="checkbox"/>	Solo orale	<input type="checkbox"/>
In caso di prova scritta i quesiti sono	A risposta multipla	<input type="checkbox"/>	A risposta libera	<input checked="" type="checkbox"/>	Esercizi numerici	<input checked="" type="checkbox"/>
Altro						

**SCHEDA DELL' INSEGNAMENTO DI *Calcolo Parallelo e Distribuito(mod. A)***

**TITOLO INSEGNAMENTO IN INGLESE *Parallel and Distributed Computing***

**Docenti: Giuliano Laccetti**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X		X		X

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Fornire idee di base, metodologie, strumenti software per lo sviluppo di algoritmi in ambiente di calcolo paralleli e/o distribuiti ad alte prestazioni. Parte integrante del corso è l'attività di laboratorio.

**CONTENUTI**

Concetto di "parallelismo" e di "alte prestazioni". I supercomputer. Classificazione e principali caratteristiche funzionali delle architetture parallele (classificazione di Flynn, rivista e aggiornata). Parametri di valutazione delle prestazioni degli algoritmi paralleli. I parametri classici di SpeedUp ed Efficiency. Metodologie per lo sviluppo di algoritmi paralleli e loro dipendenza dall'architettura. Esempi di progettazione e implementazione di algoritmi su architetture di tipo MIMD distributed memory (uso di message programming; la libreria MPI) e di tipo MIMD shared memory (l'esempio dei multicore; condivisione di memoria; la libreria OpenMP). Parametri di valutazione e scalabilità degli algoritmi paralleli. SpeedUp scalato ed Efficiency scalata. Il bilanciamento del Carico. Algoritmi tolleranti alla latenza e ai guasti. I/O parallelo. Algoritmi di base in ambiente parallelo e distribuito: ordinamenti, calcolo matriciale.

**MODALITA' DIDATTICHE**

Parte integrante del corso sono le attività di laboratorio. Durante il corso, presentazione a scadenza fissata di 2-3 miniprogetti, da realizzare in ambiente MPI e/o OpenMP; la presentazione in tempo utile (e la sufficienza della qualità del lavoro) di tali miniprogetti, esonera gli studenti dalla prova d'esame al calcolatore.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	<input checked="" type="checkbox"/>	Solo scritta	<input type="checkbox"/>	Solo orale	<input type="checkbox"/>
In caso di prova scritta i quesiti sono	A risposta multipla	<input type="checkbox"/>	A risposta libera	<input checked="" type="checkbox"/>	Esercizi numerici	<input checked="" type="checkbox"/>
Altro	Sviluppo di progetti durante il corso; Prova al calcolatore					

**SCHEDA DELL' INSEGNAMENTO DI *Calcolo Scientifico(mod. A)***

**TITOLO INSEGNAMENTO IN INGLESE *Scientific Computing***

**Docenti: Eleonora Messina**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
<b>MAT/08</b>	<b>6</b>		<b>X</b>		<b>X</b>		<b>X</b>

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Approfondimento delle problematiche legate allo sviluppo, implementazione ed analisi degli algoritmi numerici per la risoluzione di problemi significativi del mondo reale. Lo studente sarà in grado di: -analizzare e confrontare i metodi in base al diverso problema applicativo da risolvere; -interpretare i risultati computazionali anche in relazione alle proprietà di consistenza, convergenza e stabilità; risolvere modelli matematici di problemi della scienza e dell'ingegneria scegliendo metodi numerici appropriati, mediante l'implementazione degli algoritmi in un opportuno ambiente di calcolo e/o l'uso di librerie di software scientifico.

**CONTENUTI**

Sistemi lineari: fattorizzazione di matrici con speciali strutture (simmetriche definite positive, a banda, sparse) e risoluzione. Problemi di minimi quadrati lineari: risoluzione delle equazioni normali. Metodi iterativi per sistemi lineari: metodi stazionari, metodo del gradiente coniugato. Equazioni e sistemi non lineari: il metodo delle iterazioni a punto fisso, il metodo di Newton e le sue varianti. Modelli matematici ed equazioni differenziali ordinarie. Soluzione numerica di problemi ai valori iniziali, con metodi ad un passo, e di problemi ai limiti, con metodi alle differenze finite. Esempi ed applicazioni.

**MODALITA' DIDATTICHE**

Lezioni frontali ed esercitazioni in laboratorio

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale		Solo scritta		Solo orale	X
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera		Esercizi numerici	
Altro	Sviluppo di progetti e prova al calcolatore					

**SCHEDA DELL' INSEGNAMENTO DI Computer Forensics**

**TITOLO INSEGNAMENTO IN INGLESE Computer Forensics**

**Docenti:**

SSD	CFU	Anno di corso (I, II o III)			Semestre (I o II)		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X			X	

**Prerequisiti: Sistemi Operativi, Reti di calcolatori I**

**OBIETTIVI FORMATIVI**

Il corso si pone l'obiettivo di far acquisire agli studenti le competenze di base nell'ambito della Computer Forensics su aspetti teorici, tecnici, metodologie e regole giuridiche alle quali deve attenersi chi opera nel settore, con illustrazione delle tecniche paradigmatiche di indagine scientifica laddove è possibile ricorrere a prove in formato digitale sia per i casi di reati strettamente informatici, sia per gli altri tipi di illeciti in cui il dato informatico può rappresentare una prova, e relativa declinazione nel contesto normativo italiano.

**PROGRAMMA**

Introduzione all'informatica forense. Elementi, ruolo, criticità e approccio metodologico dell'informatica forense. Normative e aspetti pragmatici relativi alla costruzione della prova. Aspetti legali e tecnologici relativi all'attendibilità del dato informatico e al trattamento del reperto informatico - nello specifico la disk forensics, e il trattamento dei file systems per la corretta acquisizione e la ricostruzione di informazioni. Strumenti Hardware e Software utilizzati nella digital forensics, ivi compresi la network forensics, la mobile forensics e l'embedded forensics. Metodologie per l'acquisizione di dati crittografati.

**MODALITA' DIDATTICHE**

Lezioni frontali ed esercitazioni.

**MATERIALE DIDATTICO (max 4 righe, Times New Roman 10)**

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono (*)	A risposta multipla	X	A risposta libera	X	Esercizi numerici	
Altro (es: sviluppo progetti, prova al calcolatore ...)						



**SCHEDA DELL' INSEGNAMENTO DI *Economia e Organizzazione Aziendale***

**TITOLO INSEGNAMENTO IN INGLESE *Economics***

**Docenti: Giovanni D'Alfonso**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
ING-IND/35	6		X			X	X

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Il corso ha la finalità di introdurre gli studenti del Corso di Laurea in Informatica allo studio delle problematiche economiche, organizzative e gestionali delle imprese. In particolare relativamente alle problematiche economiche, vengono forniti gli elementi relativi ai principali problemi decisionali che l'imprenditore deve affrontare (definizione del prezzo e dei volumi di vendita, dimensione dell'impresa, ottimizzazione dei costi di produzione). La conoscenza del funzionamento delle principali grandezze economiche che caratterizzano un sistema economico attraverso lo studio della Macroeconomia proietta lo studente nella conoscenza di una dimensione economica in cui l'impresa si trova ad operare. Relativamente alla organizzazione aziendale compito principale è quello di fornire allo studente, nello specifico settore del software, modelli organizzativi che caratterizzano le piccole e medie imprese.

**CONTENUTI**

La prima parte del corso fornisce la conoscenza degli elementi di Microeconomia quali la domanda individuale, la domanda di mercato, la tecnologia, la funzione di produzione e dei costi dell'impresa, il funzionamento del mercato nelle sue diverse forme. La seconda parte del corso fornisce la conoscenza di un modello semplificato di funzionamento di un sistema economico attraverso la conoscenza dei principali elementi che caratterizzano un sistema economico (il Prodotto Nazionale, i consumi, il risparmio, l'investimento, la moneta, l'inflazione, ecc.). La terza parte del corso fornisce la conoscenza del funzionamento di una impresa sin dalla sua costituzione anche attraverso la lettura ed interpretazione dei documenti contabili e fornirà un ulteriore arricchimento del funzionamento dei meccanismi che regolano la nascita, lo sviluppo e la decadenza delle imprese. Nel corso delle lezioni vengono proposte applicazioni ed esemplificazioni dei temi trattati.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	
Altro						

**SCHEDA DELL' INSEGNAMENTO DI *Elementi di Informatica Teorica***

**TITOLO INSEGNAMENTO IN INGLESE *Introduction to Theoretical Computer Science***

**Docenti: Alessandro De Luca**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X		X		X

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Introdurre lo studente a nozioni e risultati teorici di base soggiacenti all'informatica. Lo studente potrà impadronirsi di concetti fondamentali dell'Informatica teorica e dei relativi modelli astratti di calcolo, apprezzandone l'utilità sia per un inquadramento generale del curriculum in Informatica sia per lo sviluppo delle sue capacità professionali.

**CONTENUTI**

Automi finiti e macchine sequenziali. Automi non deterministici. Linguaggi regolari. Espressioni regolari. Pumping lemma per i linguaggi regolari. Grammatiche e linguaggi indipendenti dal contesto. Forme normali di Chomski. Automi a pilae non determinismo. Corrispondenza tra automi e grammatiche. Pumping lemma per i linguaggi indipendenti dal contesto. La gerarchia di Chomsky. I concetti di algoritmo, funzione calcolabile e parzialmente calcolabile. Funzioni primitive ricorsive. La minimalizzazione. Funzioni parziali ricorsive. Numerazioni di Goedel. Macchina universale. Tesi di Church - Turing. Problemi di decisione e di enumerazione. Indecidibilità. Insiemi ricorsivi e ricorsivamente numerabili. Macchina di Turing e indecidibilità, Complessità computazionale: nozioni di base.

**MODALITA' DIDATTICHE**

Lezioni frontali ad argomento teorico ed esercitazioni per la soluzione di esercizi e problemi elementari di informatica teorica.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
Altro						

**SCHEDA DELL' INSEGNAMENTO DI *Fisica Generale I***

**TITOLO INSEGNAMENTO IN INGLESE *General Physics I***

**Docenti: Gruppo 1 Ubaldo Coscia, Gruppo 2 Francesco Tramontano**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
FIS/01	6	X				X	X

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Il contenuto del corso di Fisica è stato concepito con lo scopo preciso di far acquisire allo studente del corso di laurea in Informatica un metodo di analisi e di sintesi dei problemi da affrontare nel prosieguo dei corsi tenendo conto del metodo sperimentale proprio delle scienze fisiche. Inoltre si dà conto dei principi di base delle metodologie fisiche che potranno risultare utili al futuro laureato in Informatica.

**CONTENUTI**

1. Elementi di Meccanica ed Applicazioni: Introduzione al metodo scientifico. Cinematica del punto materiale. Principi di dinamica del punto materiale. Energia e lavoro. Dinamica dei sistemi. Corpo rigido. Gravitazione.
2. Elementi di Termodinamica: Sistemi termodinamici. Primo principio della termodinamica. Il secondo principio della termodinamica.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
------------------------------	-----------------	---	--------------	--	------------	--

In caso di prova scritta i quesiti sono	A risposta multipla	X	A risposta libera	X	Esercizi numerici	X
---	---------------------	---	-------------------	---	-------------------	---

Altro						
-------	--	--	--	--	--	--



**SCHEDA DELL' INSEGNAMENTO DI *Geometria***

**TITOLO INSEGNAMENTO IN INGLESE *Geometry***

**Docenti: Gruppo 1 Francesca Cioffi, Gruppo 2 Marco Trombetti**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
<b>MAT/03</b>	<b>6</b>	<b>X</b>				<b>X</b>	<b>X</b>

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Questo corso di primo livello ha l'obiettivo di presentare i metodi e gli strumenti dell'Algebra Lineare fornendo allo studente un approccio rigoroso allo studio dei sistemi lineari, degli spazi vettoriali, delle matrici e loro relazioni con le trasformazioni lineari, formalizzando poi la Geometria elementare del piano e dello spazio nell'ambito della teoria degli spazi vettoriali reali.

**CONTENUTI**

Conoscenza di base della teoria degli spazi vettoriali su un campo, con particolare riguardo al caso degli spazi reali di dimensione finita. Conoscenza di base del calcolo matriciale e dei legami tra matrici e trasformazioni lineari. Teoria dei sistemi lineari. Calcolo degli autovalori e degli autovettori di un operatore lineare e problema della diagonalizzazione. Piano e spazio tridimensionale euclideo, riferimenti e coordinate cartesiani e uso del linguaggio e dei metodi dell'algebra lineare per la risoluzione di problemi geometrici. Isometrie e movimenti del piano e dello spazio.

**MODALITA' DIDATTICHE**

Lezioni frontali ed esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	<input checked="" type="checkbox"/>	Solo scritta	<input type="checkbox"/>	Solo orale	<input type="checkbox"/>
In caso di prova scritta i quesiti sono	A risposta multipla	<input checked="" type="checkbox"/>	A risposta libera	<input checked="" type="checkbox"/>	Esercizi numerici	<input checked="" type="checkbox"/>
Altro						

**SCHEDA DELL' INSEGNAMENTO DI *Ingegneria del Software***

**Modulo A**

**TITOLO INSEGNAMENTO IN INGLESE *Software Engineering***

**Docenti: Sergio Di Martino**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	5			X	X		X

**Insegnamenti propedeutici previsti: Object Orientation, Algebra**

**OBIETTIVI FORMATIVI**

Ingegneria del software, dei processi di ingegneria del software e delle relative fasi, attività e deliverable (programming in the large); definizione, proprietà e analisi di modelli; metodi di analisi e progettazione (anche formali) e importanza dei linguaggi di modellazione del software per la comunicazione tra diversi attori coinvolti in un processo di ingegneria del software.

**CONTENUTI**

Introduzione all'Ingegneria del Software, concetti di prodotto software, attributi di qualità del software; L'Ingegneria dei Requisiti. Analisi e specifica dei Requisiti, Il documento dei Requisiti Software; UML: Activity Diagrams, Statecharts, Component Diagram, OCL; System Design e Architetture Software (3 Livelli, MVC, SOA, Cloud); Project Management, WBS, Diagrammi di Gantt e Pert; Versioning di Prodotti Software. SVN; Verifica e Validazione. Review, Inspection, Livelli di Testing, junit, Strategie Black Box e White Box, GUI Testing; Modelli di Processo. Il processo a cascata, i processi Agili, SCRUM.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale		Solo scritta	X	Solo orale	
------------------------------	-----------------	--	--------------	---	------------	--

In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
---	---------------------	--	-------------------	---	-------------------	---

Altro	Progetto obbligatorio di gruppo					
-------	---------------------------------	--	--	--	--	--



**SCHEDA DELL' INSEGNAMENTO DI *Ingegneria del Software***

**Modulo B**

**TITOLO INSEGNAMENTO IN INGLESE *Software Engineering***

**Docenti: Francesco Cutugno**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	5			X	X		X

**Insegnamenti propedeutici previsti: Object Orientation, Algebra**

**OBIETTIVI FORMATIVI**

User centered Design; metriche di usabilità delle interfacce, anche in ambito multimodale; definizione, proprietà e analisi di architetture software; concetti e tecniche di verifica e validazione del software.

**CONTENUTI**

Sistemi interattivi e interfacce d'uso; User Centered Design; Le regole di Usabilità di Shneiderman; I Mock-Up; Usabilità e sua valutazione; Misure di carico cognitivo e complessità delle interfacce; Grafica e metafore di rappresentazione per dati multidimensionali, Interazione multimodale.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale		Solo scritta	X	Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
Altro	Progetto obbligatorio di gruppo					

**SCHEDA DELL' INSEGNAMENTO DI *Istituzioni di Matematica 2***

**TITOLO INSEGNAMENTO IN INGLESE *Calculus II***

***Mutuato dal Corso di Laurea in Ottica e Optometria***

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
MAT/05	6		X		X		X

**Insegnamenti propedeutici previsti: Analisi Matematica I**

**OBIETTIVI FORMATIVI**

Il corso si prefigge lo scopo di introdurre gli studenti ai problemi di approssimazione di una funzione regolare mediante serie di potenze, al calcolo differenziale ed integrale per le funzioni di più variabili ed al concetto di modello matematico con particolare attenzione alle equazioni differenziali lineari.

**CONTENUTI**

Successioni e Serie di funzioni – Convergenza uniforme. Proprietà delle successioni e delle serie uniformemente convergenti. Serie totalmente convergenti. Serie di potenze: raggio di convergenza. Polinomi di Taylor: formula col resto in forma di Peano di Lagrange. Sviluppabilità in serie di Taylor: sviluppi notevoli. Cenni sulla funzione esponenziale nel campo complesso: formule di Eulero. Calcolo Differenziale – Funzioni continue, funzioni differenziabili: derivate parziali e derivate direzionali. Teorema del differenziale totale e significato geometrico. Formula di Taylor di ordine 2. Problemi di estremo libero: condizioni necessarie e condizioni sufficienti. Equazioni Differenziali – Il problema di Cauchy: Teoremi di esistenza ed unicità locale e globale. Equazioni del primo ordine a variabili separabili. Equazioni di Bernoulli. Equazioni differenziali lineari del primo e secondo ordine. Equazioni differenziali lineari del secondo ordine a coefficienti costanti, termini noti di tipo particolare. Metodo della variazione delle costanti arbitrarie. Cenni sui problemi ai limiti. Integrazione multipla – Integrale secondo Riemann. Formule di riduzione per integrali doppi e tripli. Cambiamenti di variabili in integrali doppi e tripli: il caso del cambiamento a coordinate polari.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
Altro						

**SCHEDA DELL' INSEGNAMENTO DI *Laboratorio di Algoritmi e Strutture Dati***

**TITOLO INSEGNAMENTO IN INGLESE *Algorithm and Data Structures Laboratory***

**Docenti: Gruppo 1 Fabio Mogavero; Gruppo 2 Aniello Murano**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X			X	X

**Insegnamenti propedeutici previsti: Programmazione, Laboratorio di Programmazione**

**OBIETTIVI FORMATIVI**

Obiettivo del corso è familiarizzare lo studente con la progettazione e l'implementazione di algoritmi e strutture dati. In particolare si vuole dare allo studente la capacità di produrre codice chiaro, modulare, generale ed efficiente attraverso i seguenti passi: analisi del problema, individuazione di una soluzione generale ed efficiente, stesura del codice, documentazione delle scelte effettuate e del codice prodotto.

**CONTENUTI**

Dopo un richiamo alle principali strutture dati di base, si procederà allo studio delle rappresentazioni e implementazioni di tipi di dati astratti quali Liste, Pile, Alberi e una panoramica sui Contenitori. Verrà di seguito affrontate la progettazione di librerie di base per Alberi Binari di Ricerca e Code a Priorità, che siano indipendenti dal tipo dei dati in essi contenuti (strutture dati generiche). Verranno illustrate le loro possibili implementazioni, anche in relazione alle Librerie Standard. Successivamente, si tratterà sistematicamente la rappresentazione e implementazione dei grafi e delle tecniche di visita associate. Verranno, inoltre, introdotti e implementati alcuni algoritmi che operano su grafi pesati, come ad esempio l'algoritmo di Dijkstra, quello di Bellman-Ford e tecniche di visita euristica, in particolare l'algoritmo A\*. In questo contesto, l'obiettivo generale è quello di progettare e implementare algoritmi e, più in generale, librerie che operino in maniera il più indipendente possibile dalla struttura dati concreta impiegata.

**MODALITA' DIDATTICHE**

Lezioni frontali ed esercitazioni in laboratorio.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	<input checked="" type="checkbox"/>	Solo scritta	<input type="checkbox"/>	Solo orale	<input type="checkbox"/>
In caso di prova scritta i quesiti sono	A risposta multipla	<input type="checkbox"/>	A risposta libera	<input type="checkbox"/>	Esercizi numerici	<input type="checkbox"/>
Altro	Sviluppo progetti					



**SCHEDA DELL' INSEGNAMENTO DI *Laboratorio di Programmazione***

**TITOLO INSEGNAMENTO IN INGLESE *Computer Programming Laboratory***

**Docenti: Gruppo 1 Aniello Murano; Gruppo 2 Francesco Isgrò; Gruppo 3 Roberto Prevete**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6	X				X	X

**Insegnamenti propedeutici previsti: nessuno.**

**OBIETTIVI FORMATIVI**

Il corso di Laboratorio di Programmazione ha lo scopo di fornire agli studenti le tecniche per sviluppare programmi e le prime strutture dati in linguaggio C. Il corso inizia con un riepilogo dei concetti appresi durante il corso di Programmazione I. Si porranno gli allievi di fronte a problemi di crescente complessità, che verranno risolti in modi diversi allo scopo di far apprezzare gli strumenti forniti dal linguaggio in modo efficace. Al termine del corso, gli studenti avranno acquisito le seguenti capacità operative: 1) identificazione delle strutture dati e degli algoritmi adatti alla risoluzione di semplici problemi; 2) Implementazione in linguaggio C di algoritmi e strutture dati; 3) Strutturazione modulare di un programma; 4) Utilizzo dei principali strumenti per la programmazione.

**PROGRAMMA**

Rappresentazione di dati e istruzioni. Costanti macchina. Introduzione alle funzionalità elementari del sistema operativo Unix/Linux. Linguaggio C. Tipi di dato primitivi. Input e output. Esecuzione condizionale. Iterazione. Vettori. Strutture. Funzioni. Puntatori e aritmetica dei puntatori. Strutture dati dinamiche.

Compilatore e linker. Il preprocessore del C. Debug di programmi. File header. Programmi multi-file. Ambiente di sviluppo di programmi. La documentazione del software. Le librerie standard del C.

**MODALITA' DIDATTICHE**

Lezioni frontali. Svolgimento di esercitazioni in laboratorio.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
------------------------------	-----------------	---	--------------	--	------------	--

In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera		Esercizi numerici	X
Altro	Sviluppo piccoli progetti/esercizi, algoritmi e software; prova al calcolatore					





**SCHEDA DELL' INSEGNAMENTO DI *Laboratorio di Sistemi Operativi***

**TITOLO INSEGNAMENTO IN INGLESE *Operating Systems Laboratory***

**Docenti: Gruppo 1 Marco Faella / Francesco Cutugno; Gruppo 2 Alberto Finzi /Francesco Cutugno**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	8			X	X		X

**Insegnamenti propedeutici previsti: Sistemi Operativi I, Algebra**

**OBIETTIVI FORMATIVI**

Il corso si prefigge di fornire gli strumenti e le metodologie necessarie alla gestione di sistema ed allo sviluppo di applicazioni in ambiente Unix. Al termine del corso lo studente sarà in grado di: sfruttare appieno le potenzialità di scripting per la gestione del sistema; utilizzare le interfacce di programmazione standard e delle system call al sistema Unix; progettare e realizzare programmi multi-processo e/o multi-thread; sviluppare applicazioni di rete. Inoltre, il corso si prefigge di fornire le metodologie e gli strumenti necessari per lo sviluppo di applicazioni in ambiente Android con particolare focus all'interazione tra le applicazioni ed il sistema operativo. Al termine del corso lo studente avrà appreso le problematiche relative alla creazione ed alla gestione di activity in ambienti mobile, la loro interazione con il file system presente sul dispositivo, le metodologie e gli strumenti per realizzare applicazioni multi-thread e di rete.

**CONTENUTI**

La shell di Unix: comandi e programmazione; Shell/AWK Scripting; I/O di basso livello e chiamate di sistema per la gestione l'interazione con il file system; chiamate di sistema per la gestione dei processi; chiamate di sistema per la gestione di segnali, pipe e fifo; chiamate di sistema per la creazione, gestione e sincronizzazione di thread; La comunicazione su rete. Architetture client-server; primitive di comunicazione su rete; I socket TCP ed UDP. Le chiamate di sistema per la programmazione di rete; Creazione di server concorrenti. Introduzione ai sistemi Android; Il framework Android Studio; Struttura di un progetto Android; Il ciclo di vita di una activity; Creazione e gestione di activity; Interazione con il file system; Cenni all'utilizzo di SQLite; Connessione ad Internet; Applicazioni di rete; Background processing; Multi-threading; Gestione della sincronizzazione; Servizi; Fragments; Receiver in background.

**MODALITA' DIDATTICHE**

Lezioni frontali, esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	
Altro	Sviluppo progetti					



**SCHEDA DELL' INSEGNAMENTO DI *Linguaggi di Programmazione I***

**TITOLO INSEGNAMENTO IN INGLESE *Programming Languages I***

**Docenti: Piero Bonatti**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X			X	X

**Insegnamenti propedeutici previsti: Programmazione, Laboratorio di Programmazione**

**OBIETTIVI FORMATIVI**

Fornire gli elementi tecnici per classificare i numerosissimi linguaggi di programmazione esistenti, rispetto a paradigma di computazione, caratteristiche del sistema di tipi, modalità di gestione della memoria, controllo di flusso e supporto del parallelismo. Cominciare a rendere gli studenti "utenti intelligenti" dei linguaggi di programmazione, cioè capaci di scegliere il paradigma più adatto al contesto applicativo dato, di sfruttare efficacemente le funzionalità offerte dai linguaggi e di apprendere rapidamente nuovi linguaggi. Il corso fornisce un trattamento approfondito del core di Java ed elementi di linguaggi funzionali.

**CONTENUTI**

Introduzione ai linguaggi di programmazione. Cenni storici. Richiami degli elementi informatica teorica rilevanti per il corso. Cenni ai paradigmi di programmazione. Compilazione e interpretazione dei linguaggi. Supporto a run-time e gestione della memoria. Modalità di passaggio dei parametri. Strutturazione dei dati e controllo dei tipi. Tipi elementari e user defined. Encapsulation: tipi di dato astratti, moduli, classi. Sistemi di tipo nei linguaggi ad oggetti: sottotipi ed ereditarietà; compatibilità tra tipi. Java: Costrutti di controllo e sistema di tipi in dettaglio. Tipi parametrici (programmazione generica). Strutturazione della computazione: gestione delle eccezioni. Gestione della memoria in Java (inclusi costruttori, stringhe, garbage collection e gestione dell'ambiente non locale in presenza di classi interne). Parallelismo in Java. Costrutti funzionali di base, con esempi in ML e/o in Python.

**MODALITA' DIDATTICHE**

Lezioni frontali ed esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla	X	A risposta libera	X	Esercizi numerici	
Altro						

**SCHEDA DELL' INSEGNAMENTO DI *Linguaggi di Programmazione II***

**TITOLO INSEGNAMENTO IN INGLESE *Programming Languages II***

**Docenti: Marco Faella**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X			X	X

**Insegnamenti propedeutici previsti: Linguaggi di Programmazione I, Object Orientation, Algebra**

**OBIETTIVI FORMATIVI**

Il corso intende esporre gli studenti ad un'ampia gamma di funzionalità dei moderni linguaggi di programmazione, approfondendo ed espandendo le tematiche affrontate dal corso di Linguaggi di Programmazione I, con particolare riferimento ai linguaggi orientati agli oggetti. Alla fine del corso, gli studenti saranno in grado di utilizzare i costrutti linguistici più appropriati per raggiungere gli obiettivi di chiarezza, manutenibilità, robustezza ed efficienza dei manufatti software.

**PROGRAMMA**

1. Richiami di linguaggi Java e C++.
2. Classi interne, locali e anonime.
3. Tipi enumerati.
4. Algoritmi di risoluzione del binding dinamico.
5. Operazioni di base su oggetti: confronto ordinale e di uguaglianza, copia.
6. Polimorfismo parametrico e sue implementazioni: i generics e i template.
7. Programmare con collezioni ed iteratori.
8. Programmazione multi-thread: comunicazione e sincronizzazione tra thread.
9. Riflessione.

**MODALITA' DIDATTICHE**

Lezioni frontali ed esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla	X	A risposta libera	X	Esercizi numerici	
Altro						

**SCHEDA DELL' INSEGNAMENTO DI Logica**

**TITOLO INSEGNAMENTO IN INGLESE Logic**

**Docenti: Guglielmo Tamburrini**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
M-FIL/02	6		X		X		X

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Acquisire una conoscenza delle principali proprietà sintattiche e semantiche della logica classica proposizionale e della logica del primo ordine. Acquisire familiarità con i principali sistemi deduttivi della logica classica che sono di interesse per l'informatica. Acquisire la capacità di formalizzare enunciati dichiarativi e problemi nel linguaggio della logica classica, nonché di verificare la correttezza di un ragionamento informale.

**CONTENUTI**

Logica proposizionale: sintassi e semantica. Forme normali congiuntiva e disgiuntiva. La deduzione naturale. Calcolo dei sequenti. Tableaux analitici. Risoluzione, procedura di Davis-Putnam e metodo refutazionale. Correttezza, completezza e compattezza della logica proposizionale. Logica del primo ordine: elementi di sintassi e di semantica tarskiana. Tableaux analitici. Universo di Herbrand, clausole ground e metodo refutazionale. Formalizzazione e verifica formale di ragionamenti informali. Forma normale prenessa e skolemizzazione. Correttezza, completezza e compattezza della logica del primo ordine. Teorema di Skolem-Lowenheim e modelli non-standard. Cenni ai teoremi di incompletezza di Goedel. Dimostrabilità, verità e insiemi ricorsivamente enumerabili.

**MODALITA' DIDATTICHE**

Lezioni frontali ad argomento teorico ed esercitazioni per la soluzione di esercizi e problemi elementari di logica.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
Altro						

**SCHEDA DELL' INSEGNAMENTO DI Programmazione**

**TITOLO INSEGNAMENTO IN INGLESE Computer Programming**

**Docenti: Gruppo 1 Giuliano Laccetti; Gruppo 2 Daniel Riccio**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	9	X			X		X

**Insegnamenti propedeutici previsti: nessuno**

**OBIETTIVI FORMATIVI**

Presentare il paradigma della programmazione imperativa utilizzando iterazione e ricorsione e, partendo da semplici esercizi, mettere in grado gli studenti di scrivere algoritmi non troppo complessi.

**CONTENUTI**

Introduzione al concetto di algoritmo. Rappresentazione di dati e istruzioni. Progettazione top down di un algoritmo e sua implementazione. I costrutti di controllo: sequenza, iterazione e selezione. Le functions e l'astrazione procedurale. Passaggio di parametri. Concetto di ADT. ADT Array e Stringhe. I file di testo. Algoritmi di ricerca di elemento in array. Algoritmi di ricerca di elemento in array ordinati (binary search). Algoritmi di calcolo matriciale.

ADT record. I tipi di dati derivati. La ricorsione. Utilità e potenza della ricorsione in alcuni casi significativi. Puntatori e variabili dinamiche. ADT linked list; stack; queue. Principali algoritmi per la loro gestione.

Introduzione al linguaggio C. La documentazione del software.

**MODALITA' DIDATTICHE**

Sviluppo e presentazione di piccoli progetti/esercizi durante il corso, linguaggio C

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	<input checked="" type="checkbox"/>	Solo scritta	<input type="checkbox"/>	Solo orale	<input type="checkbox"/>
In caso di prova scritta i quesiti sono	A risposta multipla	<input type="checkbox"/>	A risposta libera	<input checked="" type="checkbox"/>	Esercizi numerici	<input checked="" type="checkbox"/>
Altro	Sviluppo piccoli progetti/esercizi, algoritmi e software					

**SCHEDA DELL' INSEGNAMENTO DI *Object Orientation***

**TITOLO INSEGNAMENTO IN INGLESE *Object Orientation***

**Docenti: Sergio Di Martino**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X		X		X

**Insegnamenti propedeutici previsti: Programmazione**

**OBIETTIVI FORMATIVI**

Acquisizione delle competenze di base per la progettazione object-oriented attraverso la comprensione dei concetti di astrazione sui dati, di incapsulamento dell'informazione, di coesione e accoppiamento, e di riutilizzo del codice; comprensione delle differenze tra paradigma object-oriented e il paradigma procedurale, conoscenza del linguaggio java per la definizione di classi e per la promozione del riutilizzo del software capacità di applicare conoscenza e comprensione delle principali abilità (ossia la capacità di applicare le conoscenze acquisite) saranno: analisi di problemi, specifica dei requisiti e definizione di una strategia risolutiva con un approccio orientato agli oggetti, con la sua implementazione nel linguaggio java, garantendo il giusto equilibrio tra qualità ed efficienza del software.

**CONTENUTI**

La programmazione orientata agli oggetti; concetti di astrazione dei dati e di incapsulamento; Progettazione di classi. Concetti di coesione e accoppiamento; Ereditarietà e riuso; Interfacce, classi astratte e polimorfismo; UML: Class Diagrams e Sequence Diagrams; Introduzione a Java, alla JVM e al JDK; Oggetti, variabili, riferimenti; classi e metodi, costruttori, comunicazione fra oggetti, parametri espliciti e impliciti; il riferimento this; Tipi di dati fondamentali. Classi Object e String; Meccanismi di 'autoboxing' e 'unboxing'; Accenni di gestione delle Eccezioni; Le collezioni in Java: List e sue implementazioni. For generico (for each); Design pattern: Iterator, Observer, Strategy, Composite; Programmazione di interfacce grafiche ad eventi. Le Swing.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale		Solo scritta	X	Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
Altro	Progetto					



**SCHEDA DELL' INSEGNAMENTO DI Reti di Calcolatori I**

**TITOLO INSEGNAMENTO IN INGLESE Computer Network I**

**Docenti: Guido Russo**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6			X	X		X

**Insegnamenti propedeutici previsti: Sistemi operativi I, Algebra**

**OBIETTIVI FORMATIVI**

L'obiettivo del Corso è quello di introdurre i concetti fondamentali delle moderne reti di calcolatori e fornire allo studente le necessarie conoscenze per affrontare l'analisi e lo studio di una rete distribuita di calcolatori. In particolare, saranno presentate le caratteristiche generali delle reti, la loro topologia, l'architettura ed i principali protocolli utilizzati per la trasmissione delle informazioni tra calcolatori, con particolare riferimento ai protocolli della suite TCP/IP ed ai moderni apparati attivi di rete.

**CONTENUTI**

- Introduzione alle reti di calcolatori – Scopi, applicazioni, evoluzione storica, caratteristiche e struttura a livelli.
  - Il modello ISO/OSI – Il modello di riferimento, protocolli, interfacce, principali architetture di rete, le primitive.
  - Il livello fisico - Mezzi trasmissivi elettrici e ottici. Codifica e decodifica del segnale. Tecniche di multiplexazione. Elementi di Cablaggio strutturato. Reti in rame e fibra. Repeater.
  - Il livello Data Link- Il modello di riferimento IEEE 802. La rete Ethernet e lo standard IEEE 802.3: metodo di accesso CSMA/CD. Apparati attivi: bridge, switch L3. Lo standard 802.11: metodo di accesso CSMA/CA.
  - Il livello di rete – I protocolli della suite TCP/IP. Repeater, bridge, switch L3, router, gateway). Tecniche di instradamento, architettura di un router, algoritmi di routing.
  - Il livello di trasporto - Servizi forniti, indirizzi del livello di trasporto, le socket. I protocolli UDP e TCP.
  - Il livello delle applicazioni: Applicazioni di Rete: Domain Name Server (DNS). Protocolli del livello applicativo: SMTP, POP3, IMAP, SNMP, TELNET, FTP, TFTP, DHCP, NAT.
  - Progetto di una rete di calcolatori. La progettazione di una rete LAN e WAN. 4
- Sicurezza nelle reti: la sicurezza nella posta: SSL, TLS; nell'autenticazione: 802.1X, EAP; nelle sessioni interattive: SSH. 4

**MODALITA' DIDATTICHE**

Il corso prevede delle lezioni frontali in aula, delle esercitazioni in aula svolte dal docente, delle simulazioni di esame scritto. Numerosi esempi reali di reti e di configurazioni di apparati di rete. Visita al Data Center SCOPE/RECAS e analisi della sua rete.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	
Altro	2 esercizi, 1 di teoria ed 1 di progettazione rete					

**SCHEDA DELL' INSEGNAMENTO DI Reti di calcolatori II**

**TITOLO INSEGNAMENTO IN INGLESE Computer network II**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X			X	X

**Insegnamenti propedeutici previsti: Reti di calcolatori I**

**OBIETTIVI FORMATIVI**

Il corso intende fornire una visione specialistica dei sistemi di comunicazione in generale e delle reti di calcolatori, del loro funzionamento e delle loro applicazioni, con particolare riferimento alle tecnologie di rete locale, metropolitana e geografica, alla loro interconnessione ed a tutte le problematiche di conduzione ed esercizio di infrastrutture di rete complesse. Saranno illustrate in profondità le tecnologie alla base dell'erogazione dei principali servizi di rete, la loro gestione nei principali ambienti operativi ed applicazioni nella rete Internet. In particolare l'attività formativa sarà orientata alla presentazione delle nozioni teoriche alla base della trasmissione dei dati, delle problematiche di commutazione ad alte prestazioni e dell'internet working e all'approfondimento delle principali tecnologie di rete.

**CONTENUTI**

Introduzione. Caratteristiche di una rete di calcolatori, tipi di reti, aspetti progettuali, gli standard. La comunicazione e la trasmissione delle informazioni: concetti generali. Tecnologie per la trasmissione: tecniche di moltiplicazione a divisione di tempo e di frequenza, commutazione di circuito e commutazione di pacchetto, comunicazione in rete connection oriented e connectionless. I mezzi trasmissivi: la trasmissione delle informazioni - fondamenti teorici. Modelli multilayer di riferimento per la strutturazione delle reti: modello ISO/OSI - architetture a livelli, interfacce fra livelli, incapsulamento, primitive di colloquio, modalità peer-to-peer e host-to-host. L'architettura di rete TCP/IP: architettura, protocollo IP, indirizzamento IP e instradamento, il protocollo TCP. Scalabilità e gestione avanzata di reti LAN, multilayer switching e Qualità del Servizio. Elementi e problematiche di configurazione avanzata e troubleshooting di apparecchiature e di interfacce LAN, elementi di Network Management. Principali tecnologie WAN: Collegamenti di terra e satellitari. Elementi e problematiche di configurazione di interfacce WAN. Routing: problematiche avanzate di instradamento e gestione degli spazi di indirizzamento, evoluzione del concetto di routing e forwarding/switching, strategie e protocolli di routing. IGP routing: routing statico e dinamico - Protocolli distance vector (RIP v1 e v2) e link-state (OSPF). Elementi di configurazione e troubleshooting dei protocolli di routing IGP. EGP routing: la struttura tier-based della rete internet, il concetto di Autonomous System, problematiche e politiche di gestione e smistamento del traffico. Il transito. Il protocollo BGP 4. Elementi di configurazione e troubleshooting di rete WLAN (IEEE802.11, ecc.) e WMAN (Hiperlan, ecc.).

**MODALITA' DIDATTICHE**

Lezioni frontali con esercitazioni

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale		Solo scritta		Solo orale	X
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	
Altro	Elaborato su argomento concordato.					

**SCHEDA DELL' INSEGNAMENTO DI *Ricerca Operativa***

**TITOLO INSEGNAMENTO IN INGLESE *Operations Research***

**Docenti: Paola Festa**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
<b>MAT/09</b>	<b>6</b>		<b>X</b>		<b>X</b>		<b>X</b>

**Insegnamenti propedeutici previsti: Algoritmi e strutture dati I**

**OBIETTIVI FORMATIVI**

L'insegnamento si prefigge quale obiettivo principale l'introduzione degli studenti all'uso dei modelli di programmazione matematica ed in particolare ai modelli di ottimizzazione lineare (sia continui che a variabili intere) ed alle loro applicazioni nei campi della logistica, dei servizi e della produzione industriale. L'impostazione metodologica del Corso, inoltre, punta al conseguimento dei seguenti ulteriori obiettivi intermedi:

- capacità di formalizzazione dei modelli di ottimizzazione per problemi di logistica, organizzazione, pianificazione, scheduling, trasporto, flusso su reti e problemi su grafi ed alberi;
- conoscenza della teoria e dei metodi di ottimizzazione lineare continua, di ottimizzazione lineare discreta e di ottimizzazione su grafi, alberi e reti di flusso;
- capacità di utilizzazione dei modelli matematici dei classici problemi di ottimizzazione e dei relativi algoritmi di risoluzione nei campi della Pianificazione della Produzione, della Localizzazione, della Gestione delle Scorte e della Logistica.

**CONTENUTI**

Problemi di Programmazione Lineare e Metodo del Simplex. Definizione e classificazione dei problemi di ottimizzazione e dei problemi di decisione e classificazione dei relativi metodi risolutivi (metodi esatti, metodi di approssimazione e metodi euristici). Programmazione Lineare (PL): il Metodo del Simplex. Problemi di Programmazione Lineare Intera (1 credito) Metodi esatti per la risoluzione dei problemi di Programmazione Lineare Intera (Branch & Bound; piani di taglio; programmazione dinamica). Esempi di problemi di PLI con matrice dei vincoli uni-modulare: il problema del trasporto ed il problema dell'assegnamento. Problemi dello Zaino. Un algoritmo Branch and Bound per il problema dello Zaino 0/1; un algoritmo greedy per il problema dello Zaino Frazionario; due algoritmi di Programmazione Dinamica per il problema dello Zaino 0/1. Problemi di Ottimizzazione su grafi ed alberi: Vertex Cover ed Albero di Copertura Minimo. Il problema del Vertex Cover: un algoritmo 2-approssimato per il problema del Vertex Cover. Il problema dell'albero di copertura di un grafo a costo minimo (MST): l'algoritmo di Kruskal. Problemi di Ottimizzazione su grafi ed alberi: Problemi di Cammino Minimo. Cammini in un grafo orientato: il problema della raggiungibilità (visita in ampiezza; visita in profondità). Il problema dei cammini minimi: l'algoritmo di Dijkstra; l'algoritmo di Floyd e Warshall. Problemi di Ottimizzazione su grafi ed alberi: Pianificazione di un Progetto e Problema del Massimo Flusso. Pianificazione di un progetto: il Metodo CPM. Problemi di flusso su reti: il problema del massimo flusso; teorema max-flow min-cut; algoritmo di Ford-Fulkerson.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	X
Altro						

**SCHEDA DELL' INSEGNAMENTO *Sistemi informativi multimediali***

**TITOLO INSEGNAMENTO IN INGLESE *Multimedia Information Systems***

**Docenti: *Walter Balzano***

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X			X	X

**Insegnamenti propedeutici previsti: Basi di dati I**

**OBIETTIVI FORMATIVI**

Il corso tratta i principali modelli e tecniche per la gestione dei dati e dei sistemi informativi multimediali. Particolari riferimenti sono relativi ai meccanismi di storing, ricerca e browsing per contenuto su database multimediali, relazione tra database multimediali ed il Web. Particolare attenzione è rivolta a sistemi di localizzazione quali GPS, Fingerprinting ed INS.

**CONTENUTI**

Il corso è suddiviso in due parti. Prima parte: definizioni e classificazioni dei Media e dei Multimedia. Gestione di dati multimediali audio/video, dalla digitalizzazione alla consultazione degli stessi con particolari riferimenti ai concetti di Storing, Digital Signal Processing, Compressione e Streaming. Seconda parte: Sistemi Multimediali Digitali, Distribuiti ed Interattivi. Valutazioni di complessità, controllo ed adattamento. Presentazione ed interfacce utente. Cenni ai Sistemi Informativi Multimediali con riferimento alla relazione tra Database multimediali ed il Web. Sistemi di Localizzazione, GPS, Fingerprinting ed Inertial Measurement System.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	<input type="checkbox"/>	Solo scritta	<input type="checkbox"/>	Solo orale	<input checked="" type="checkbox"/>
In caso di prova scritta i quesiti sono	A risposta multipla	<input type="checkbox"/>	A risposta libera	<input type="checkbox"/>	Esercizi numerici	<input type="checkbox"/>
Altro						

**SCHEDA DELL' INSEGNAMENTO Sistemi Operativi I**

**TITOLO INSEGNAMENTO IN INGLESE Operating System I**

**Docenti: Walter Balzano**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	9		X			X	X

**Insegnamenti propedeutici previsti: Architettura degli Elaboratori**

**OBIETTIVI FORMATIVI**

Il corso intende fornire una introduzione alla struttura e alle funzioni dei moderni Sistemi Operativi esaminandone i principi, le componenti fondamentali, le metodologie di progettazione e di sviluppo, gli algoritmi e gli strumenti di base. Particolari riferimenti riguardano il Sistema Operativo Unix ed implementazioni Linux, conoscenza delle metodologie usate per risolvere le problematiche tipiche della gestione delle risorse. Ulteriore finalità del corso è quella di fornire abilità di base nell'uso di una piattaforma a livello utente ed amministratore, principi di scripting e programmazione di Sistema.

**PROGRAMMA**

Introduzione ai Sistemi Operativi: Definizioni di strutture, architetture e componenti.

Gestione dei processi: Concetti, operazioni e comunicazioni sui processi; Definizioni di Thread; Gestione della CPU, criteri ed algoritmi di Scheduling e valutazione degli Algoritmi. Scheduling per sistemi di elaborazione in tempo reale. Sincronizzazione dei processi; Problema della sezione critica; Architetture di sincronizzazione; Semafori; Problemi tipici di sincronizzazione; Regioni critiche; Monitor. Stallo dei Processi: Rilevamento e ripristino da situazioni di stallo.

Gestione della memoria: Avvicendamento dei processi, assegnazione contigua della memoria; Paginazione; Segmentazione. La memoria virtuale, definizioni di paginazione e segmentazione. Interfaccia e realizzazione del file system, Concetto di file, metodi di accesso, condivisione e protezione di file. Realizzazione della directory, metodi di assegnazione, gestione dello spazio libero, efficienza e prestazioni.

Sistemi di I/O: Architetture e dispositivi di I/O; Interfaccia di I/O per le applicazioni; Sottosistema per l'I/O del nucleo; Trasformazione delle richieste di I/O in operazioni dei dispositivi; Prestazioni. Memoria secondaria e terziaria, struttura dei dischi, scheduling del disco, gestione dell'unità a disco, Gestione dell'area di avvicendamento; Strutture RAID; Connessione dei dischi; Strutture per la memorizzazione terziaria.

Cenni alle architetture distribuite riguardanti aspetti di strutture, di protezione e sicurezza.

**MODALITA' DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla	X	A risposta libera	X	Esercizi numerici	X
Altro						



**SCHEDA DELL' INSEGNAMENTO DI *Sistemi Operativi II***

**TITOLO INSEGNAMENTO IN INGLESE *Operating Systems II***

**Docenti: Silvio Barra**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6		X			X	X

**Insegnamenti propedeutici previsti: Sistemi Operativi I**

**OBIETTIVI FORMATIVI**

Il corso si pone come obiettivo principale quello di analizzare in modo approfondito e dettagliato gli algoritmi e le strutture dati implementati in un sistema operativo (Linux 2.6). In secondo luogo esso affronta le stesse problematiche, in contesti differenti, quali i dispositivi mobile ed il Cloud.

**CONTENUTI**

Il corso di Sistemi Operativi II ha una duplice finalità. In primo luogo, esso intende completare lo studio dei sistemi operativi tradizionali affrontati nel corso di Sistemi Operativi 1, approfondendo alcuni concetti come la gestione della memoria, dei processi e degli interrupt in un sistema operativo specifico, ossia Linux con kernel 2.6. D'altro canto approfondisce i medesimi aspetti nel contesto dei sistemi operativi mobile con particolare attenzione a MAC iOS e Android. L'ultima parte del corso è invece dedicata all'approfondimento di tematiche legate ai sistemi operativi real-time e Cloud. Nello specifico, i principali temi affrontati riguardano: 1) I Sistemi Operativi Open-Source Linux OS, 2) La Gestione della Memoria in Linux, 3) La Gestione dei Processi in Linux, 4) Lo Scheduling dei Processi, 5) I Processi e la Memoria, 6) Il Virtual File System, 7) I sistemi Grid, Cloud e WebOS, 8) Dispositivi e Sistemi Mobile, 9) Symbian OS, 10) Android OS, 11) Mac iOS, 12) La sicurezza nei sistemi operativi mobile (SELinux vs. SEAndroid), 13) Accesso sicuro ad un dispositivo mobile e/o tramite un dispositivo mobile, 14) I sistemi operativi Real-time.

**MODALITA' DIDATTICHE**

Lezioni frontali.

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale		Solo scritta		Solo orale	X
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera		Esercizi numerici	
Altro						



**SCHEDA DELL' INSEGNAMENTO DI *Tecnologie Web***

**TITOLO INSEGNAMENTO IN INGLESE *Web Technologies***

**Docenti: Anna Corazza**

SSD	CFU	Anno di corso			Semestre		Lingua
		I	II	III	I	II	Italiano
INF/01	6			X	X		X

**Insegnamenti propedeutici previsti: Linguaggi di Programmazione I, Object Orientation, Algebra**

**OBIETTIVI FORMATIVI**

Scopo del corso è di fornire concetti e tecniche per la progettazione di sistemi web. Alla fine del corso, lo studente dovrebbe quindi essere in grado di progettare un'applicazione web scegliendo gli strumenti più adatti e di seguire l'evoluzione delle tecnologie legate a questo campo di applicazione.

**CONTENUTI**

Introduzione al web: protocollo HTTP, architettura a tre e più strati, web statico e web dinamico, il web come ipertesto e come grafo, linguaggi di mark-up. Session tracking. Programmazione lato server: CGI; strumenti basati su Java; cenni di PHP. Programmazione lato client: JavaScript, fogli di stile (CSS e XSL), DOM, AJAX. Servizi web. Introduzione alla programmazione su web (bots, spiders e crawlers). Frameworks e CMS.

**MODALITA' DIDATTICHE**

Lezioni frontali.

**MATERIALE DIDATTICO**

**MODALITA' DI ESAME**

L'esame si articola in prova	Scritta e orale	X	Solo scritta		Solo orale	
In caso di prova scritta i quesiti sono	A risposta multipla		A risposta libera	X	Esercizi numerici	
Altro						