



UNIVERSITÀ DEGLI STUDI  
DI NAPOLI FEDERICO II

SCUOLA POLITECNICA E DELLE SCIENZE DI BASE  
DIPARTIMENTO DI INGEGNERIA ELETTRICA E TECNOLOGIE  
DELL'INFORMAZIONE

# **CORSO DI LAUREA IN INFORMATICA**

*Classe delle Lauree in Scienze e Tecnologie Informatiche - L31*

## **GUIDA DELLO STUDENTE**

ANNO ACCADEMICO 2026/2027

Vers. 1.0

AGGIORNATA: 17/07/2026

## 1 Sommario

|           |                                                                                                        |           |
|-----------|--------------------------------------------------------------------------------------------------------|-----------|
| <b>1</b>  | <b>Sommario</b>                                                                                        | <b>2</b>  |
| <b>2</b>  | <b>Revisioni</b>                                                                                       | <b>5</b>  |
| <b>3</b>  | <b>Informazioni Generali sul Corso di Laurea Triennale in Informatica</b>                              | <b>6</b>  |
| <b>4</b>  | <b>Quadro Normativo degli Studi Universitari</b>                                                       | <b>7</b>  |
|           | <b>Come iscriversi</b>                                                                                 | <b>8</b>  |
|           | Test di ingresso                                                                                       | 8         |
|           | <b>Obiettivi formativi</b>                                                                             | <b>9</b>  |
|           | <b>Condizione occupazionale dei laureati</b>                                                           | <b>11</b> |
| <b>5</b>  | <b>Regolamento vigente</b>                                                                             | <b>12</b> |
| <b>6</b>  | <b>Percorso didattico</b>                                                                              | <b>13</b> |
|           | Percorso didattico – Regolamento dal 2024/25                                                           | 13        |
|           | Percorso didattico – Regolamento dal 2018/19 al 2023/24                                                | 13        |
| <b>7</b>  | <b>Manifesto degli Studi del Corso di Laurea in Informatica</b>                                        | <b>14</b> |
|           | Manifesto degli Studi (per studenti immatricolati dall'a.a. 2024/25)                                   | 15        |
|           | Manifesto degli Studi (per studenti immatricolati da a.a. 2018/19 a 2023/24)                           | 17        |
| <b>8</b>  | <b>Transizione dal precedente regolamento al nuovo regolamento</b>                                     | <b>19</b> |
| <b>9</b>  | <b>Piani di studio ed Esami a Libera Scelta</b>                                                        | <b>20</b> |
| <b>10</b> | <b>Tirocinio</b>                                                                                       | <b>21</b> |
|           | Riconoscimento di crediti per attività formative svolte esternamente al Corso di Studi (e.g.: Academy) | 22        |
|           | Riconoscimento di crediti per attività lavorativa                                                      | 22        |
| <b>11</b> | <b>Esame finale e attribuzione del punteggio finale</b>                                                | <b>23</b> |
| <b>12</b> | <b>Calendario Attività Didattiche, Orario delle Lezioni e Calendario Esami</b>                         | <b>25</b> |
|           | Calendario delle Attività didattiche per l'a.a. 2026/2027                                              | 25        |
|           | Orario delle lezioni                                                                                   | 25        |
|           | Calendario degli Esami di profitto                                                                     | 25        |
|           | Calendario degli Esami di Laurea                                                                       | 25        |
| <b>13</b> | <b>Suddivisione in canali</b>                                                                          | <b>26</b> |
| <b>14</b> | <b>Referenti del Corso di Studi</b>                                                                    | <b>27</b> |
|           | Coordinatore Didattico dei Corsi di Studio in Informatica                                              | 27        |
|           | Referenti per le attività di tirocinio                                                                 | 27        |

|                                                                |           |
|----------------------------------------------------------------|-----------|
| <b>Referente del Corso di Studi per il Programma ERASMUS+:</b> | <b>27</b> |
| <b>Referente Orientamento</b>                                  | <b>27</b> |
| <b>Responsabile dei Contenuti del Sito Web</b>                 | <b>27</b> |
| <b>Referente Pratiche Studenti</b>                             | <b>27</b> |
| <b>Rappresentanti degli Studenti</b>                           | <b>27</b> |
| <b>15 Vademecum Studenti DIETI</b>                             | <b>28</b> |
| <b>16 Contatti e Strutture</b>                                 | <b>29</b> |
| <b>17 Schede degli insegnamenti</b>                            | <b>30</b> |
| <b>Schede degli Insegnamenti – Manifesto a.a. 2026-2027</b>    | <b>31</b> |
| Primo Anno                                                     | 31        |
| Programmazione                                                 | 31        |
| Architettura degli Elaboratori                                 | 33        |
| Analisi Matematica I                                           | 34        |
| Programmazione Object-Oriented                                 | 35        |
| Basi di dati                                                   | 36        |
| Algebra                                                        | 37        |
| Secondo Anno                                                   | 38        |
| Analisi e progettazione di Algoritmi                           | 38        |
| Analisi e progettazione di Strutture Dati                      | 39        |
| Elementi di Informatica Teorica                                | 40        |
| Fisica Generale I                                              | 41        |
| Geometria                                                      | 42        |
| Linguaggi di Programmazione                                    | 43        |
| Metodi Statistici per l'informazione                           | 44        |
| Sistemi Operativi                                              | 45        |
| Terzo Anno                                                     | 46        |
| AI Technologies                                                | 46        |
| Reti e Programmazione Distribuita                              | 47        |
| Tecniche di Programmazione Avanzata                            | 49        |
| Tecnologie Web                                                 | 50        |
| Ingegneria del Software                                        | 52        |
| Esami a Scelta Libera                                          | 54        |
| Algorithm design                                               | 54        |
| Calcolo Numerico                                               | 55        |
| Diritto dell'Informatica                                       | 56        |
| Computer Forensics                                             | 58        |
| Economia e Organizzazione Aziendale                            | 59        |
| Istituzioni di Matematica II                                   | 60        |
| Logic for computer science                                     | 61        |
| Operating systems for mobile, cloud and IoT                    | 62        |
| Parallel and Distributed Computing                             | 63        |
| Operation Research                                             | 64        |
| Scientific Computing                                           | 65        |
| Multimedia Information Systems                                 | 66        |

|                                                                             |           |
|-----------------------------------------------------------------------------|-----------|
| <b>Schede degli Insegnamenti – Manifesto ante 2024-2025</b>                 | <b>67</b> |
| Primo Anno                                                                  | 67        |
| Algebra (DISATTIVATO DALL'A.A. 2024/2025)                                   | 67        |
| Analisi Matematica I                                                        | 69        |
| Architettura degli Elaboratori                                              | 70        |
| Fisica Generale I                                                           | 71        |
| Geometria                                                                   | 72        |
| Laboratorio di Programmazione (DISATTIVATO DALL'A.A. 2024/2025)             | 73        |
| Programmazione (DISATTIVATO DALL'A.A. 2024/2025)                            | 74        |
| Secondo Anno                                                                | 75        |
| Algoritmi e Strutture Dati I (DISATTIVATO DALL'A.A. 2025/2026)              | 75        |
| Basi di dati I (DISATTIVATO DALL'A.A. 2025/2026)                            | 77        |
| Elementi di Informatica Teorica                                             | 79        |
| Laboratorio di Algoritmi e Strutture Dati (DISATTIVATO DALL'A.A. 2025/2026) | 80        |
| Linguaggi di Programmazione I (DISATTIVATO DALL'A.A. 2025/2026)             | 81        |
| Object Orientation (DISATTIVATO DALL'A.A. 2025/2026)                        | 82        |
| Sistemi Operativi I (DISATTIVATO DALL'A.A. 2025/2026)                       | 83        |
| Terzo Anno                                                                  | 85        |
| Laboratorio di Sistemi Operativi (DISATTIVATO DALL'A.A. 2026/2027)          | 85        |
| Calcolo delle Probabilità e Statistica (DISATTIVATO DALL'A.A. 2025/2026)    | 87        |
| Ingegneria del Software Modulo A (DISATTIVATO DALL'A.A. 2026/2027)          | 89        |
| Ingegneria del Software Modulo B (DISATTIVATO DALL'A.A. 2026/2027)          | 90        |
| Reti di Calcolatori I (DISATTIVATO DALL'A.A. 2026/2027)                     | 91        |
| Esami a Scelta Vincolata (Tab. A)                                           | 93        |
| Linguaggi di Programmazione II (DISATTIVATO DALL'A.A. 2026/2027)            | 93        |
| Tecnologie Web (DISATTIVATO DALL'A.A. 2026/2027)                            | 95        |
| Esami a Scelta Libera (Tab. B)                                              | 97        |
| Algorithm design                                                            | 97        |
| Calcolo Numerico                                                            | 99        |
| Diritto dell'Informatica                                                    | 100       |
| Computer Forensics                                                          | 101       |
| Economia e Organizzazione Aziendale                                         | 102       |
| Istituzioni di Matematica II                                                | 103       |
| Logic for computer science                                                  | 104       |
| Operating systems for mobile, cloud and IoT                                 | 105       |
| Parallel and Distributed Computing                                          | 106       |
| Ricerca Operativa                                                           | 107       |
| Scientific Computing                                                        | 109       |
| Multimedia Information Systems                                              | 110       |

## 2 Revisioni

Questo documento potrebbe subire modifiche volte a migliorarne l'organizzazione e/o a correggere eventuali refusi. Gli studenti sono invitati ad assicurarsi di consultare sempre l'ultima versione disponibile. La versione più aggiornata del documento è consultabile alla [pagina dedicata sul sito web del Corso di Studi](#).

| Data       | Versione | Descrizione                           |
|------------|----------|---------------------------------------|
| 30/06/2026 | 1.0      | Prima versione della guida 2026/2027. |

### 3 Informazioni Generali sul Corso di Laurea Triennale in Informatica

Il Corso di Laurea Triennale in Informatica dell'Università di Napoli Federico II, classe L-31, forma da oltre 25 anni professionisti in grado di progettare, sviluppare e gestire sistemi informatici e soluzioni digitali, con una crescente attenzione alle soluzioni basate sull'intelligenza artificiale.

Il percorso offre una preparazione solida e completa: integra le basi scientifiche e matematiche dell'informatica con attività pratiche, esercitazioni, progetti e laboratori. Gli studenti acquisiscono competenze nella programmazione, nella progettazione di algoritmi e strutture dati, nella gestione di basi di dati e nella comprensione dei sistemi operativi e delle reti. Vengono inoltre introdotte le principali tecnologie emergenti, come l'intelligenza artificiale, e i loro ambiti applicativi, fornendo una visione sempre aggiornata dell'evoluzione del settore.

Il corso è pensato sia per chi desidera entrare rapidamente nel mondo del lavoro, sia per chi intende proseguire gli studi con la Laurea Magistrale in Informatica, approfondendo ulteriormente i temi dell'intelligenza artificiale e delle soluzioni informatiche più avanzate. Le competenze acquisite consentono di inserirsi in team di sviluppo software, analisi dei dati, progettazione di sistemi intelligenti e servizi digitali, oppure di intraprendere un percorso professionale autonomo.

Una parte fondamentale della del percorso di formazione è il tirocinio, che si svolge generalmente presso aziende, enti o laboratori di ricerca. Il Corso di Laurea mantiene rapporti con una vasta rete di realtà professionali, offrendo agli studenti l'opportunità di confrontarsi con problemi concreti e con contesti in cui l'intelligenza artificiale e le tecnologie digitali giocano un ruolo sempre più centrale.

**Il laureato in Informatica**, previo superamento dell'Esame di Stato, può inoltre iscriversi all'Albo degli Ingegneri dell'Informazione, sezione B, ottenendo il **titolo di Ingegnere dell'Informazione Junior**.

I Corsi di Laurea e di Laurea Magistrale in Informatica sono inquadrati nel Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione (DIETI). Il DIETI ha ottenuto dal Ministero della Ricerca il riconoscimento di **Dipartimento di Eccellenza** per il quinquennio 2023/2027.

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

## 4 Quadro Normativo degli Studi Universitari

L'attuale quadro normativo italiano sull'organizzazione degli studi universitari, in ottemperanza al modello europeo, prevede:

- Un titolo di studio di primo livello (denominato **Laurea**) della durata di 3 anni.
- Un titolo di studio di secondo livello (denominato **Laurea Magistrale**) della durata di 2 anni.

Coloro che conseguono la Laurea di primo livello possono proseguire gli studi ed ottenere, mediamente dopo altri due anni, il titolo di studio di secondo livello (Laurea Magistrale).

Un concetto rilevante è quello di **Credito Formativo Universitario (CFU)**, che rappresenta l'unità di misura del lavoro di apprendimento dello studente. Ad 1 CFU corrispondono mediamente 25 ore di lavoro di apprendimento che possono comprendere ore di: lezioni frontali, laboratorio, seminari, tirocinio e cultura di contesto, studio individuale, etc.

Per ogni esame superato lo studente acquisisce il numero di CFU assegnati al relativo insegnamento. Viene convenzionalmente previsto un lavoro di apprendimento annuale pari all'acquisizione di 60 CFU. Pertanto, la Laurea di primo livello viene conseguita dopo l'acquisizione di 180 CFU, mentre la Laurea Magistrale richiede l'acquisizione di ulteriori 120 CFU.

## Come iscriversi

Il Corso di Laurea in Informatica è ad **accesso libero**: non è quindi prevista una prova selettiva per l'ammissione.

Per immatricolarsi è tuttavia necessario sostenere il test di ingresso obbligatorio **TOLC-I**, previsto per i Corsi di Laurea in Ingegneria dell'Università di Napoli Federico II. Il test ha finalità di **autovalutazione** e si basa su un questionario a risposta multipla. Il mancato superamento del test non impedisce l'iscrizione al Corso di Laurea, ma può comportare l'attribuzione di Obblighi Formativi Aggiuntivi, secondo le modalità indicate nella sezione successiva.

L'immatricolazione si effettua attraverso [SegrePass](#), il sistema di Ateneo che permette di accedere ai servizi per gli studenti. L'immatricolazione e l'iscrizione agli anni successivi hanno luogo, di norma, dal 16 luglio al 31 ottobre di ogni anno, con modalità che sono rese note sul [sito web di Ateneo](#). Per ulteriori informazioni è possibile consultare la [Guida alle Immatricolazioni e Test di Ingresso](#) disponibile sul sito web del corso di Studi in Informatica.

## Test di ingresso

L'immatricolazione al Corso di Laurea in Informatica prevede un Test di autovalutazione obbligatorio. Le modalità di iscrizione al test sono quelle indicate nel [sito web](#) della Scuola Politecnica e delle Scienze di Base. In particolare, il documento che regola le modalità e il calendario dei test di ammissione si trova alla seguente [pagina](#) (Test di ammissione "on line" (**TOLC-I**) per Corsi di Studio in Ingegneria e in Scienze **non a numero programmato**).

Le modalità per prenotare il test per il Corso di Laurea in Informatica sono quelle dei **Corsi di Studio in Ingegneria e in Scienze non a numero programmato**.

È possibile l'iscrizione al corso di laurea anche senza aver fatto ancora il test di autovalutazione (si intende che lo studente dovrà affrontare il test nelle date previste successive all'immatricolazione).

La soglia di superamento è di **15/50 per la parte logico-matematica e di 15 per la conoscenza della lingua inglese**.

Il superamento della soglia di conoscenza della lingua inglese permette l'acquisizione automatica di 3 CFU per "Conoscenza della lingua inglese".

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 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 l'attribuzione di un **Obbligo Formativo Aggiuntivo (OFA)** di 3 CFU, che è estinto sostenendo con esito positivo, **prima di ogni altro esame**, un esame a scelta tra Analisi matematica I, Architettura degli Elaboratori, o Programmazione.

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:

- capacità di interpretare il significato di un testo e di sintetizzarlo o di rielaborarlo in forma scritta e orale;
- abilità di comprendere e rispondere a quesiti attenendosi strettamente agli elementi forniti;
- capacità di individuare i dati di un problema pratico e di utilizzarli per pervenire alla risoluzione nella maniera più rapida;
- 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.

Non sono richieste conoscenze pregresse di informatica.

## Obiettivi formativi

Il Corso di Laurea in Informatica ha l'obiettivo di formare laureati con solide competenze nei principali ambiti delle scienze e tecnologie dell'informazione. Il percorso prepara gli studenti sia all'uso consapevole dei sistemi informatici, sia alla loro progettazione, realizzazione e gestione.

La formazione combina conoscenze teoriche, metodologiche e applicative. Gli studenti acquisiscono le basi dell'informatica, della matematica e della fisica necessarie per comprendere il funzionamento dei sistemi di elaborazione, sviluppare software, analizzare problemi complessi e progettare soluzioni informatiche efficaci.

In coerenza con le linee guida nazionali e internazionali del settore, tra cui quelle del GRIN e dell'ACM, il percorso didattico prevede l'acquisizione di competenze nei seguenti ambiti:

- fondamenti di matematica, matematica discreta, matematica del continuo e fisica;
- architetture dei sistemi di elaborazione e reti di comunicazione;
- sistemi operativi e gestione delle risorse di calcolo;
- algoritmi, strutture dati e analisi della complessità;
- metodologie di programmazione e principali paradigmi dei linguaggi di programmazione;
- progettazione, gestione e interrogazione di basi di dati;
- progettazione e realizzazione di sistemi software;
- conoscenze affini e interdisciplinari utili alla comprensione dei contesti applicativi dell'informatica.

Il percorso dà particolare rilievo anche alle attività pratiche, progettuali e laboratoriali, che consentono agli studenti di applicare le conoscenze acquisite a problemi concreti. La preparazione si completa con il tirocinio formativo, svolto presso aziende, enti, laboratori di ricerca, strutture della pubblica amministrazione o, in alternativa, attraverso attività progettuali coerenti con gli obiettivi del Corso di Studi.

Si prevede anche l'incentivazione di soggiorni di studio presso università straniere, nel quadro di accordi internazionali, quali ad esempio il programma europeo Erasmus+.

### **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 carattere pratico/laboratoriale previste per i principali settori dell'attività formativa caratterizzante (Programmazione, Programmazione Object-Oriented, Analisi e Progettazione di Strutture Dati) e, più in generale, nelle attività di progettazione ed esercitazioni di gruppo svolte nell'ambito dei principali insegnamenti caratterizzanti. Ulteriori competenze specifiche vengono acquisite durante l'attività di tirocinio finale.

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

### **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;

## Condizione occupazionale dei laureati

La condizione occupazionale dei laureati è rilevata attraverso le indagini indipendenti di *AlmaLaurea*<sup>1</sup>, che indicano che la Laurea Triennale in Informatica dell'Università degli Studi di Napoli Federico II **garantisce un ottimo livello occupazionale**, superiore alla media nazionale relativa ai CdL della stessa classe. Per i laureati considerati nella più recente **Indagine 2022**, il tasso di disoccupazione def. Istat è dello **0%**. Il tempo medio che intercorre tra la laurea e il primo lavoro è di soli 1,8 mesi. La percezione dell'efficacia della laurea nel lavoro svolto nell'indagine per i laureati del 2022 è positiva per circa il **91%** degli intervistati. La **retribuzione media** è superiore alla media nazionale (1.499€ mensili per i laureati in Informatica dell'Ateneo Federico II, a fronte di 1.426€ nella media nazionale).

---

<sup>1</sup> I dati delle indagini sono consultabili pubblicamente su sito [www.alma laurea.it](http://www.alma laurea.it)

## 5 Regolamento vigente

A partire dall'anno accademico 2024/2025, il Corso di Laurea Triennale in Informatica ha adottato una variazione di regolamento che ne ha aggiornato sia la struttura che i contenuti. Questo aggiornamento è stato implementato con l'obiettivo di migliorare l'offerta formativa, adattandola alle esigenze del mercato del lavoro e alle innovazioni tecnologiche.

È importante che ciascuno studente faccia riferimento al regolamento in vigore nell'anno della propria immatricolazione. Pertanto:

- Gli studenti immatricolati dall'anno accademico 2024/2025 seguono il nuovo regolamento attualmente vigente.
- Gli studenti immatricolati negli anni accademici precedenti seguono il piano di studi articolato secondo il regolamento vigente nell'anno della loro immatricolazione.

I regolamenti del Corso di Studi sono disponibili all'indirizzo:

➤ <https://informatica.dieti.unina.it/index.php/it/laurea-triennale/regolamento-laurea-triennale>

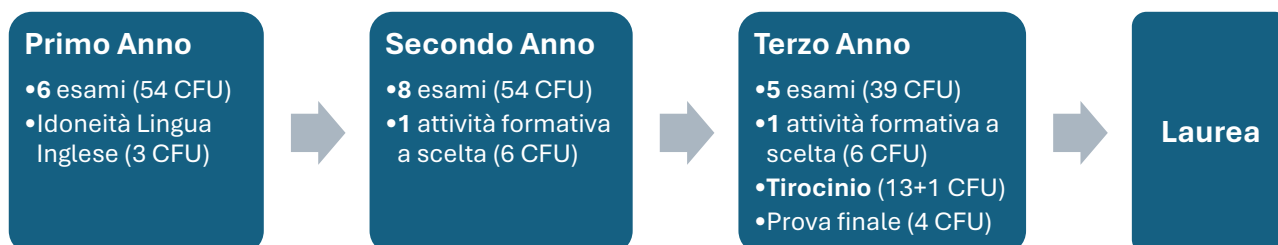
Questa guida è organizzata per fornire a tutti gli studenti le informazioni necessarie per comprendere il proprio percorso di studi. Per garantire chiarezza e facilità di consultazione, la guida include sezioni apposite dedicate ai piani di studi e alle schede degli insegnamenti per gli ultimi due regolamenti in vigore. Ove non diversamente specificato, le informazioni riportate in questa guida valgono indipendentemente da regolamento vigente al momento dell'immatricolazione.

## 6 Percorso didattico

Il percorso didattico per il conseguimento della Laurea Triennale in Informatica è articolato in tre anni.

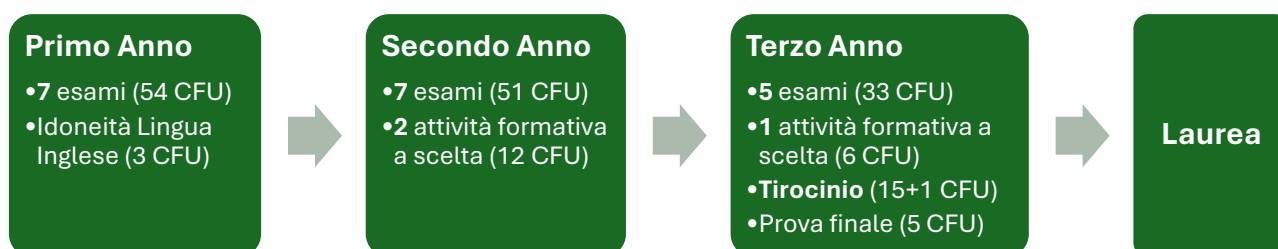
### Percorso didattico – Regolamento dal 2024/25

Per gli studenti immatricolati dall'a.a. 2024/2025, il percorso didattico previsto è riportato nella figura seguente.



### Percorso didattico – Regolamento dal 2018/19 al 2023/24

Per gli studenti immatricolati negli anni accademici dal 2018/2019 al 2023/2024, il percorso didattico previsto è riportato nella figura seguente.



Per gli studenti immatricolati in anni accademici antecedenti il 2018/2019, si rimanda ai regolamenti specifici, consultabili sul [sito web del Corso di Studi](#).

## 7 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 della sua immatricolazione.

Nelle sezioni seguenti è riportato il Manifesto degli Studi approvato dall'a.a. 2024/2025 (riferimento per gli immatricolati nell'a.a. 2026/27) e per gli anni precedenti.

- [Manifesto degli Studi \(per studenti immatricolati dall'a.a. 2024/2025\)](#)
- [Manifesto degli Studi \(per studenti immatricolati dall'a.a. 2018/19 a 2023/24\)](#)

Ciascuno studente deve fare riferimento al Manifesto degli Studi in vigore nell'anno in cui si è immatricolato. Gli studenti immatricolati in anni accademici antecedenti il 2018/19, possono consultare il manifesto degli studi nella guida dello studente dell'anno di immatricolazione.

## Manifesto degli Studi (per studenti immatricolati dall'a.a. 2024/25)

| Insegnamento o attività formativa                              | Semestre* | CFU | SSD       | Propedeuticità                                                                                                           |
|----------------------------------------------------------------|-----------|-----|-----------|--------------------------------------------------------------------------------------------------------------------------|
| <b>I anno</b>                                                  |           |     |           |                                                                                                                          |
| <a href="#">Analisi Matematica I</a>                           | 1         | 9   | MATH-03/A |                                                                                                                          |
| <a href="#">Architettura degli Elaboratori</a>                 | 1         | 9   | INFO-01/A |                                                                                                                          |
| <a href="#">Programmazione</a>                                 | 1         | 12  | INFO-01/A |                                                                                                                          |
| Inglese                                                        | 1         | 3   |           |                                                                                                                          |
| <a href="#">Algebra</a>                                        | 2         | 6   | MATH-02/A |                                                                                                                          |
| <a href="#">Programmazione Object-Oriented</a>                 | 2         | 9   | INFO-01/A |                                                                                                                          |
| <a href="#">Basi di Dati</a>                                   | 2         | 9   | INFO-01/A |                                                                                                                          |
| <b>II anno</b>                                                 |           |     |           |                                                                                                                          |
| <a href="#">Analisi e Progettazione di Strutture Dati</a>      | 1         | 9   | INFO-01/A | <a href="#">Programmazione</a>                                                                                           |
| <a href="#">Fisica Generale I</a>                              | 1         | 6   | PHYS-01/A |                                                                                                                          |
| <a href="#">Elementi di Informatica Teorica</a>                | 1         | 6   | INFO-01/A |                                                                                                                          |
| <a href="#">Sistemi Operativi</a>                              | 2         | 9   | INFO-01/A | <a href="#">Architettura degli elaboratori</a>                                                                           |
| <a href="#">Geometria</a>                                      | 1         | 6   | MATH-02/B |                                                                                                                          |
| <a href="#">Analisi e Progettazione di Algoritmi</a>           | 2         | 6   | INFO-01/A | <a href="#">Analisi Matematica I</a><br><a href="#">Programmazione</a>                                                   |
| <a href="#">Linguaggi di Programmazione</a>                    | 2         | 6   | INFO-01/A | <a href="#">Programmazione</a><br><a href="#">Programmazione Object-Oriented</a>                                         |
| <a href="#">Metodi Statistici per l'Informazione</a>           | 2         | 6   | IINF-03/A | <a href="#">Analisi Matematica I</a>                                                                                     |
| Esami a libera scelta (vedi Tab. Insegnamenti a scelta libera) |           | 6   |           |                                                                                                                          |
| <b>III anno</b>                                                |           |     |           |                                                                                                                          |
| Reti e programmazione distribuita                              | 1         | 9   | INFO-01/A | <a href="#">Algebra</a>                                                                                                  |
| <a href="#">Ingegneria del Software</a>                        | 1         | 9   | INFO-01/A | <a href="#">Programmazione Object-Oriented</a><br><a href="#">Algebra</a>                                                |
| <a href="#">Tecnologie Web</a>                                 | 1         | 9   | INFO-01/A | <a href="#">Programmazione Object-Oriented</a><br><a href="#">Algebra</a>                                                |
| <a href="#">Tecniche di Programmazione Avanzata</a>            | 2         | 6   | INFO-01/A | <a href="#">Programmazione Object-Oriented</a><br><a href="#">Linguaggi di Programmazione</a><br><a href="#">Algebra</a> |
| <a href="#">AI Technologies</a>                                | 2         | 6   | INFO-01/A | <a href="#">Algebra</a>                                                                                                  |
| Esami a libera scelta (vedi Tab. Insegnamenti a scelta libera) |           | 6   | INFO-01/A |                                                                                                                          |
| Tirocinio                                                      |           | 13  |           |                                                                                                                          |
| Altre attività di orientamento                                 |           | 1   |           |                                                                                                                          |
| Prova finale                                                   |           | 4   |           |                                                                                                                          |

**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)**

| Insegnamenti a scelta libera                                | SSD       | C<br>F<br>U | Semestre | Propedeuticità                                                                    |
|-------------------------------------------------------------|-----------|-------------|----------|-----------------------------------------------------------------------------------|
| <a href="#">Algorithm design</a>                            | INFO-01/A | 6           | 2        | Analisi e Progettazione di Strutture Dati<br>Analisi e Progettazione di Algoritmi |
| <a href="#">Istituzioni di matematica II</a>                | MATH-03/A | 6           | 1        | <a href="#">Analisi matematica I</a>                                              |
| <a href="#">Calcolo numerico</a>                            | MATH-05/A | 6           | 1        |                                                                                   |
| <a href="#">Parallel and distributed computing</a>          | INFO-01/A | 6           | 1        |                                                                                   |
| <a href="#">Scientific computing</a>                        | MATH-05/A | 6           | 1        |                                                                                   |
| <a href="#">Computer Forensics</a>                          | INFO-01/A | 6           | 2        |                                                                                   |
| Diritto dell'Informatica                                    | GIUR-17/A | 6           | 2        |                                                                                   |
| <a href="#">Economia ed organizzazione aziendale</a>        | IEGE-01/A | 6           | 2        |                                                                                   |
| <a href="#">Logic for computer science</a>                  | INFO-01/A | 6           | 2        |                                                                                   |
| <a href="#">Operation Research</a>                          | MATH-06/A | 6           | 1        | Analisi e Progettazione di Strutture Dati                                         |
| <a href="#">Multimedia Information Systems</a>              | INFO-01/A | 6           | 2        |                                                                                   |
| <a href="#">Operating systems for mobile, cloud and IoT</a> | INFO-01/A | 6           | 1        | Sistemi Operativi                                                                 |

## Manifesto degli Studi (per studenti immatricolati da a.a. 2018/19 a 2023/24)

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 compresi tra l'a.a. 2018/19 e l'a.a. 2023/24.

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

| Materie a scelta Tabella A                     | SSD    | CFU | Semestre | Propedeuticità                                                 |
|------------------------------------------------|--------|-----|----------|----------------------------------------------------------------|
| <a href="#">Linguaggi di programmazione II</a> | INF/01 | 6   | 2        | Algebra<br>Linguaggi di programmazione I<br>Object Orientation |
| <a href="#">Tecnologie web</a>                 | INF/01 | 6   | 2        | Algebra<br>Linguaggi di programmazione I<br>Object Orientation |

**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                                                                                                                            |
|-------------------------------------------------------------|------------|-----|----------|-------------------------------------------------------------------------------------------------------------------------------------------|
| <a href="#">Algorithm design</a>                            | INF/01     | 6   | 2        | <a href="#">Algoritmi e Strutture Dati I</a><br><a href="#">Laboratorio di Algoritmi e Strutture Dati</a>                                 |
| <a href="#">Istituzioni di matematica II</a>                | MAT/05     | 6   | 1        | <a href="#">Analisi matematica I</a>                                                                                                      |
| <a href="#">Calcolo numerico</a>                            | MAT/08     | 6   | 2        |                                                                                                                                           |
| <a href="#">Parallel and distributed computing</a>          | INF/01     | 6   | 1        |                                                                                                                                           |
| <a href="#">Scientific computing</a>                        | MAT/08     | 6   | 1        |                                                                                                                                           |
| <a href="#">Computer Forensics</a>                          | INF/01     | 6   | 2        |                                                                                                                                           |
| Diritto dell'Informatica                                    | IUS/20     | 6   | 2        |                                                                                                                                           |
| <a href="#">Economia ed organizzazione aziendale</a>        | ING-IND/35 | 6   | 2        |                                                                                                                                           |
| <a href="#">Linguaggi di programmazione II</a>              | INF/01     | 6   | 2        | <a href="#">Object orientation</a><br><a href="#">Linguaggi di programmazione I</a><br><a href="#">Algebra</a>                            |
| <a href="#">Logic for computer science</a>                  | M-FIL/02   | 6   | 2        |                                                                                                                                           |
| <a href="#">Operation Research</a>                          | MAT/09     | 6   | 1        | <a href="#">Algoritmi e strutture dati I</a>                                                                                              |
| <a href="#">Multimedia Information Systems</a>              | INF/01     | 6   | 2        |                                                                                                                                           |
| <a href="#">Operating systems for mobile, cloud and IoT</a> | INF/01     | 6   | 2        | <a href="#">Sistemi Operativi I</a>                                                                                                       |
| <a href="#">Tecnologie web</a>                              | INF/01     | 6   | 2        | <a href="#">Object orientation</a><br><a href="#">Linguaggi di programmazione I</a><br><a href="#">Algebra</a><br><a href="#">Algebra</a> |
| AI Technologies                                             | INFO-01/A  | 6   | 2        |                                                                                                                                           |

## 8 Transizione dal precedente regolamento al nuovo regolamento

Con l'anno accademico 2026/27 si conclude la transizione al nuovo regolamento, con l'attivazione dei nuovi insegnamenti del terzo anno, come segue:

➤ **a.a. 2026/2027:**

- Attivazione degli insegnamenti del terzo anno secondo il nuovo regolamento, tra cui “AI Technologies”, Tecnologie Web da 9 CFU, Tecniche di Programmazione Avanzata, Ingegneria del Software da 9 CFU e Reti e Programmazione Distribuita.
- Disattivazione degli insegnamenti del terzo anno secondo il precedente regolamento (e.g.: Ingegneria del Software da 10 CFU, Tecnologie Web da 6 CFU, Linguaggi di Programmazione II, Reti di Calcolatori, Laboratorio di Sistemi Operativi).

**In ogni caso, gli studenti immatricolati con il regolamento ante 2024/25 potranno continuare a sostenere gli esami presenti nel loro piano di studi seguendo i programmi in vigore al momento della loro immatricolazione, anche quando gli insegnamenti saranno disattivati. Per gli insegnamenti disattivati, non si terranno più lezioni frontali.**

## 9 Piani di studio ed Esami a Libera Scelta

I crediti legati agli esami a libera scelta possono essere conseguiti scegliendo liberamente tra tutti gli insegnamenti attivati presso l'Ateneo che siano coerenti con il percorso formativo della Laurea Triennale in Informatica.

- Se tali crediti 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, e può sostenere direttamente gli esami a scelta libera.
- Se lo studente intende conseguire, anche solo parzialmente, tali CFU con insegnamenti erogati dall'Ateneo ma non attivati per il Corso di Laurea in Informatica (i.e., non presenti in Tabella B), allora è necessario presentare un piano di studi al Coordinatore del Corso di Studi **entro il 31 ottobre 2026 (inviando una e-mail al coordinatore del corso di studi con indicazione della scelta)**.

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 crediti 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).

## 10 Tirocinio

Il Corso di Studi Triennale in Informatica prevede, nell'Articolo 17 del Regolamento Didattico, lo svolgimento di un tirocinio curriculare, o attività equivalente, connesso al completamento della preparazione culturale degli studenti.

Le attività di Tirocinio sono disciplinate dal Regolamento (Decreto interministeriale n.142 del 25/03/98) recante le norme di attuazione dei principi e dei criteri di cui all'art.18 della legge 24 giugno 1997, n.196 (cd. legge Treu), sui tirocini formativi e di orientamento. Il Tirocinio rappresenta un momento di alternanza tra studio e lavoro che si svolge nell'ambito dei processi formativi, con lo scopo di agevolare le scelte professionali dello studente, favorendone la conoscenza diretta del mondo del lavoro.

Il tirocinio ha, dunque, per lo studente una duplice valenza:

- **formativa**, perché mira ad approfondire ed ampliare le conoscenze acquisite durante il percorso di studi;
- **orientativa**, perché fornisce l'opportunità di una conoscenza diretta della realtà professionale.

Il tirocinio non è un rapporto di lavoro e non dà alcun diritto di natura economica al tirocinante. L'Università si fa carico degli oneri assicurativi per gli infortuni, per le malattie professionali, per la responsabilità civile nonché delle comunicazioni obbligatorie alle Organizzazioni sindacali e agli Uffici preposti ai controlli in materia di lavoro, purché il tirocinio si tenga sul territorio europeo. In caso di tirocini svolti al di fuori del territorio europeo, sarà l'azienda a farsi carico dell'assicurazione.

Il tirocinio si può svolgere presso un **Soggetto Ospitante** (azienda, ente pubblico o privato, e istituzione di ricerca, ...) **convenzionato** con l'Università degli Studi di Napoli Federico II (tirocinio *extra moenia* o esterno), oppure presso l'Università degli Studi di Napoli Federico II (tirocinio *intra moenia* o interno). In entrambi i casi, è necessario che l'addestramento conseguente sia rivolto a una maturazione del candidato, sia culturale che professionale.

È fortemente consigliato che il tirocinio venga svolto presso aziende esterne, o 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 curricolari. Gli studenti possono svolgere il tirocinio anche presso istituzioni di ricerca come CNR, INFN, INFN, Osservatorio Astronomico ed altri, oppure 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, l'impegno richiesto per portare a termine il tirocinio è determinato dal numero di crediti corrispondente nel piano di studi dello studente. Per ogni credito formativo è previsto un impegno nelle attività di tirocinio pari a 25 ore. Per gli studenti immatricolati dall'a.a. 2024/2025, il piano di studi prevede **13 CFU** di tirocinio, corrispondenti a un impegno di  $13 \times 25 = \mathbf{325 \text{ ore}}$ . Per gli studenti immatricolati in anni accademici precedenti, il piano di studi prevede un tirocinio da 15 CFU, corrispondente a  $15 \times 25 = 375 \text{ ore}$ .

Le modalità di svolgimento del tirocinio curricolare, inclusi i requisiti per l'avvio del tirocinio e l'iter burocratico da seguire, sono disciplinati dalla Commissione per il Coordinamento Didattico del Corso di Studi in Informatica nell'apposito Regolamento Tirocini.

Informazioni dettagliate sui tirocini e sull'iter per l'assegnazione di un tirocinio sono disponibili sul sito web del Corso di Studi, alle pagine riportate di seguito:

- Il **Regolamento Tirocini** è consultabile sul sito web del Corso di Studi, al seguente indirizzo: <https://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/tirocini/regolamento-tirocini>
- L'**Iter Burocratico** da seguire e i requisiti per richiedere l'assegnazione del tirocinio sono disponibili nella pagina web dedicata sul sito del Corso di Studi, all'indirizzo: <https://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/tirocini/iter-burocratico>
- Un elenco (parziale) di aziende con offerte per tirocini esterni è disponibile sul sito web del Corso di Studi, all'indirizzo: <https://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/tirocini/aziende-con-offerte-per-tirocini>

### Riconoscimento di crediti per attività formative svolte esternamente al Corso di Studi (e.g.: Academy)

Come indicato nel Regolamento Tirocini (Art. 10), cui si rimanda per ulteriori dettagli, gli studenti che abbiano completato percorsi in Academy organizzate dall'Università degli Studi di Napoli Federico II su tematiche attinenti insegnamenti caratterizzanti del Corso di Studi possono richiedere il riconoscimento di al più **3 CFU** di Tirocinio.

I restanti crediti di Tirocinio andranno acquisiti con un tirocinio intra moenia o extra moenia, secondo le procedure descritte nel Regolamento Tirocini.

Lo studente può sottoporre la domanda per tale riconoscimento alla Commissione Tirocini, allegando adeguata documentazione (e.g.: attestato di completamento del percorso in Academy). L'elenco delle Academy per cui è previsto il riconoscimento automatico dei crediti è stabilito con provvedimento della CCD e pubblicato sul sito web del Corso di Studi in Informatica. Nel caso di completamento di più di una Academy, vengono riconosciuti comunque al massimo 3 CFU complessivamente.

Per richiedere un eventuale riconoscimento di crediti per altre attività equiparabili a tirocinio curricolare non incluse tra quelle con automatico riconoscimento, lo studente sottopone al Coordinatore del Corso di Studi adeguata documentazione sulle attività formative e professionali svolte.

### Riconoscimento di crediti per attività lavorativa

Come indicato nel Regolamento Tirocini (Art. 10-bis), cui si rimanda per ulteriori dettagli, gli studenti che abbiano recentemente svolto un'attività lavorativa su tematiche attinenti agli insegnamenti caratterizzanti del Corso di Studi e agli obiettivi formativi del tirocinio possono richiedere il riconoscimento dei CFU previsti per il tirocinio curricolare, secondo le modalità dettagliate nel Regolamento Tirocini.

Per ulteriori dettagli, si rimanda al Regolamento Tirocini e all'Iter Burocratico per tirocini dettagliato sul sito web del Corso di Studi.

## 11 Esame finale e attribuzione del punteggio finale

L'esame finale di laurea consiste nella redazione di un **elaborato di tesi**, tipicamente sviluppato a partire dall'esperienza del tirocinio (interno o esterno), e nella presentazione dello stesso alla Commissione di Laurea.

L'elaborato di tesi è un documento redatto dal candidato sotto la guida del Relatore assegnato dalla Commissione Tirocini, che per prassi coincide con il Tutor Accademico che ha seguito il tirocinio. Non sono previsti formati specifici per gli elaborati di tesi.

La presentazione dell'elaborato di tesi ha una durata massima di 10 minuti. Durante la presentazione, gli studenti possono avvalersi di supporti audio-visivi (e.g.: presentazione realizzata con Microsoft PowerPoint o strumenti simili). Al termine della presentazione, i membri della Commissione di Laurea possono porre domande ai candidati sul lavoro svolto.

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

1. Il voto medio degli esami del candidato, espresso in centodecimi;
2. Il punteggio espresso dalla Commissione di Laurea in seguito alla prova del candidato, compreso tra 1 e 6 punti;
3. Il punteggio di **incentivazione per la velocità**, compreso tra 0 e 5 punti, definito come segue:
  - a. **5 punti**, se il candidato si laurea entro la fine del terzo anno accademico dalla sua immatricolazione.
  - b. **2 punti**, se il candidato si laurea entro la fine del quarto anno accademico dalla sua immatricolazione.
  - c. **0 punti**, altrimenti.
4. Il punteggio di **incentivazione per la media**, compreso tra 0 e 4 punti, definito come segue:
  - a. Se il voto medio in centodecimi  $X$  degli esami del candidato è uguale o superiore a 81, il punteggio di incentivazione si calcola secondo la seguente formula:  $X * 3/22 - 11; 0$  punti, altrimenti.

Qualora il voto medio degli esami del candidato, espresso in trentesimi, sia maggiore o uguale di 28 e la valutazione finale sia massima (110/110), la Commissione di Laurea può concedere la lode, con decisione unanime.

La richiesta di lode va comunicata dal relatore alla Commissione di Laurea almeno una settimana prima della data dell'esame finale, ove si ritenga che il candidato ne possieda i requisiti.

## 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.

Informazioni più dettagliate possono essere recuperate nel sito del Corso di Studi all'indirizzo:

- <https://informatica.dieti.unina.it/index.php/it/servizi-per-gli-studenti/erasmus>

Il [referente del Programma Erasmus+](#) del Corso di Studi è il Prof. Luigi Sauro.

## 12 Calendario Attività Didattiche, Orario delle Lezioni e Calendario Esami

Questa sezione riporta dettagli sul calendario delle attività didattiche, sull'orario delle lezioni e sul calendario degli esami e delle sedute di laurea.

### Calendario delle Attività didattiche per l'a.a. 2026/2027

Il calendario delle attività didattiche è stabilito dalla Scuola Politecnica e delle Scienze di Base ed è consultabile alla [pagina dedicata sul sito web della Scuola](#).

Per l'a.a. 2026/2027, le lezioni del primo periodo didattico (primo semestre) si terranno orientativamente **dal 14 settembre al 18 dicembre 2026**. Le lezioni del secondo periodo didattico (secondo semestre) si terranno orientativamente **dal 1 marzo all'11 giugno 2027**.

**Le date precise saranno pubblicate non appena disponibili.**

Nei periodi di esami, sono garantiti cinque appelli ordinari e due appelli di recupero (come definiti nel Regolamento Didattico di Ateneo), definiti come segue:

- Appelli ordinari obbligatori: un appello nel mese di gennaio 2027, un appello nel mese di febbraio 2027, un appello nel mese di giugno 2027, un appello nel mese di luglio 2027 ed un appello nel mese di settembre 2027.
- Appelli di recupero obbligatori: un appello nel mese di marzo 2027, un appello nel periodo ottobre–novembre 2027.
  - **Le date precise saranno pubblicate non appena disponibili.**

### Orario delle lezioni

L'orario delle lezioni è consultabile on-line all'indirizzo:

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

### Calendario degli Esami di profitto

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

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

### Calendario degli Esami di Laurea

Il calendario degli Esami di Laurea è disponibile sul sito web della Scuola Politecnica e delle Scienze di Base, al seguente indirizzo:

- <http://www.scuolapsb.unina.it/index.php/laurea-ingegneria>

Le Commissioni degli Esami di Laurea e l'elenco dei candidati sono consultabili on-line all'indirizzo:

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

## 13 Suddivisione in canali

Nell'anno accademico 2026/27 i corsi della laurea triennale sono suddivisi in base al cognome in canali secondo la seguente modalità:

**Primo Anno** (lezioni a [Monte Sant'Angelo](#))

- **Canale 1: Cognomi lettere A-DE**
- **Canale 2: Cognomi lettere DF-M**
- **Canale 3: Cognomi lettere N-Z**

**Secondo Anno** (lezioni a [Monte Sant'Angelo](#))

- **Canale 1: Cognomi lettere A-G**
- **Canale 2: Cognomi lettere H-Z**

**Terzo Anno** (lezioni a [via Claudio e Piazzale Tecchio](#)/[Monte Sant'Angelo](#))

- **Canale 1: Cognomi lettere A-G**
- **Canale 2: Cognomi lettere H-Z**

**Si sottolinea che gli studenti non possono in alcun modo cambiare il proprio canale di appartenenza, e sono vincolati a sostenere l'esame con il docente del proprio canale.**

## 14 Referenti del Corso di Studi

In questa pagina sono riportati i nomi e i riferimenti del coordinatore del Corso di Studi in Informatica e dei docenti referenti per diversi aspetti specifici. In calce, sono riportati anche i nomi e gli indirizzi e-mail dei rappresentanti degli studenti per il Corso di Studi Triennale.

### Coordinatore Didattico dei Corsi di Studio in Informatica

- Prof. Sergio Di Martino - tel. 081/679272 - e-mail: [cds.informatica@unina.it](mailto:cds.informatica@unina.it)

### Referenti Tirocini e Riconoscimento Crediti Lavorativi

- Prof. Massimo Benerecetti – [massimo.benerecetti@unina.it](mailto:massimo.benerecetti@unina.it)
- Prof. Alessandro De Luca – [alessandro.deluca@unina.it](mailto:alessandro.deluca@unina.it).
- Prof. Alberto Finzi – [alberto.finzi@unina.it](mailto:alberto.finzi@unina.it)
- Prof. Fabio Mogavero – [fabio.mogavero@unina.it](mailto:fabio.mogavero@unina.it).

### Referente del Corso di Studi per il Programma ERASMUS+:

- Prof. Luigi Sauro – [luigi.sauro@unina.it](mailto:luigi.sauro@unina.it).

### Referente Orientamento

- Prof. Daniel Riccio – [daniel.riccio@unina.it](mailto:daniel.riccio@unina.it)

### Responsabile dei Contenuti del Sito Web e Social Media

- Prof. Luigi Libero Lucio Starace – [luigiliberolucio.starace@unina.it](mailto:luigiliberolucio.starace@unina.it)

### Referenti Pratiche studenti, piani di studio, e acquisizione crediti di altre attività formative

- Prof. Francesco Isgrò – [francesco.isgro@unina.it](mailto:francesco.isgro@unina.it)
- Prof. Riccardo Caccavale – [riccardo.caccavale@unina.it](mailto:riccardo.caccavale@unina.it)
- Prof. Alessandra Rossi – [alessandra.rossi@unina.it](mailto:alessandra.rossi@unina.it)

### Rappresentanti degli Studenti

I nomi e gli indirizzi email dei rappresentanti degli studenti sono riportati di seguito e elencati nella [pagina apposita](#) sul sito web del Corso di Studi.

- Di Tota Gaetano – [g.ditota@studenti.unina.it](mailto:g.ditota@studenti.unina.it)
- Sorrentino Myriam – [my.sorrentino@studenti.unina.it](mailto:my.sorrentino@studenti.unina.it)
- Palmieri Emanuele Gerardo – [emanu.palmieri@studenti.unina.it](mailto:emanu.palmieri@studenti.unina.it)
- Tibello Francesco – [f.tibello@studenti.unina.it](mailto:f.tibello@studenti.unina.it)
- Anepeta Alessio – [a.anepeta@studenti.unina.it](mailto:a.anepeta@studenti.unina.it)
- Ciotola Emanuele – [emanu.ciotola@studenti.unina.it](mailto:emanu.ciotola@studenti.unina.it)

## 15 Vademecum Studenti DIETI

Il Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione, cui il Corso di Studi in Informatica afferisce, ha redatto un Vademecum per gli studenti, per orientarsi nelle strutture e nei servizi erogati. Il documento è reperibile al seguente indirizzo:

- <https://www.dieti.unina.it/index.php/it/studenti/vademecum-dello-studente-menu>

## 16 Contatti e Strutture

I corsi del primo e secondo anno sono erogati nel plesso di [Monte Sant'Angelo, Via Vicinale Cupa Cintia, 26, 80126 Napoli NA](#).

I corsi del terzo anno sono erogati nel plesso di [Via Claudio, 21, 80125 Napoli NA](#) / [Piazzale Vincenzo Tecchio, 80138, Napoli NA](#) (Canale 1) e [Monte Sant'Angelo, Via Vicinale Cupa Cintia, 26, 80126 Napoli NA](#) (Canale 2).

### Siti web utili:

- Sito web del Corso di Studio: <https://informatica.dieti.unina.it/>
- Sito web del Dipartimento: <https://dieti.unina.it>
- Sito web della Scuola: <http://www.scuolapsb.unina.it>
- Sito web di Ateneo: [www.unina.it](http://www.unina.it)
- Portale Orientamento: [www.orientamento.unina.it](http://www.orientamento.unina.it)

### Canali Social ufficiali:

- Instagram: <https://www.instagram.com/unina.informatica/>
- LinkedIn: <https://www.linkedin.com/company/informatica-federico-ii/>
- Facebook: <https://www.facebook.com/InformaticaUnina/>
- Canale Telegram Ufficiale: [https://t.me/informatica\\_unina](https://t.me/informatica_unina)

## 17 Schede degli insegnamenti

In questa sezione sono riportate le Schede degli Insegnamenti attivati per il Regolamento in vigore dall'a.a. 2024/2025 e per il regolamento precedente.

Le Schede degli Insegnamenti riportano, per ciascun insegnamento, docenti titolari della cattedra (eventualmente per i diversi canali), obiettivi formativi, contenuti del corso, eventuali propedeuticità, modalità d'esame e materiale didattico. Per ulteriori dettagli, gli studenti possono consultare le schede degli insegnamenti inserite da tutti i docenti sulla piattaforma <https://www.docenti.unina.it> e le schede degli insegnamenti riportate nell'Allegato 2.1 del [Regolamento Didattico del Corso di Studi](#).

## Schede degli Insegnamenti – Manifesto a.a. 2026-2027

### Primo Anno

#### Programmazione

##### TITOLO INSEGNAMENTO IN INGLESE *Computer Programming*

|                |                                                                                            |
|----------------|--------------------------------------------------------------------------------------------|
| SSD            | INFO-01/A                                                                                  |
| CFU            | 12                                                                                         |
| Anno           | Primo                                                                                      |
| Semestre       | Primo                                                                                      |
| Propedeuticità | Nessuna                                                                                    |
| Docenti        | Canale 1: Roberto Prevete; Canale 2: Daniel Riccio/Luigi Sauro; Canale 3: Francesco Isgrò. |

#### OBIETTIVI FORMATIVI

L'obiettivo del Corso è quello di 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. Il corso ha anche lo scopo di fornire le competenze pratiche per sviluppare programmi e le prime strutture dati in linguaggio C. Gli studenti saranno in grado di utilizzare ambienti di sviluppo a riga di comando e IDE.

#### 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. Algoritmi di ricerca di elemento in array. Algoritmi di ricerca di elemento in array ordinati (binary search). Algoritmi di calcolo matriciale. Algoritmi di ordinamento. ADT record. I tipi di dati derivati. La ricorsione. Utilità e potenza della ricorsione in alcuni casi significativi. Puntatori e variabili dinamiche. Allocazione dinamica della memoria. ADT linked list; stack; queue e principali algoritmi per la loro gestione. Introduzione al linguaggio C. La documentazione del software

#### Esercizi

- Esercizi di programmazione per ogni argomento del corso
- Progetti di programmazione di complessità crescente

#### Laboratorio:

- Introduzione a bash, al compilatore gcc e all'editor testuale nano.
- Riepilogo delle istruzioni di controllo e indentazione di un programma C.
- Scrivere programmi sintatticamente corretti: Errori tipici di sintassi; come interpretare i messaggi di errore del compilatore; buone pratiche per evitare errori di sintassi e per commentare un programma.
- Il ciclo di sviluppo di un programma: design dell'algoritmo, design della sessione di testing, implementazione, debugging. Esempi di sviluppo di un programma C.
- Esprimere condizioni complesse mediante espressioni booleane. Pattern condizionali su array: esistenziale, universale, minimo, massimo.
- Tecniche di sviluppo un po' più avanzate: interpretare i warning di gcc; utilizzo del debugger gdb; compilazione modulare.
- Analisi delle prestazioni di algoritmo, caso di studio con algoritmi di ordinamento. Opzioni di compilazione di gcc per aumentare le prestazioni.
- Funzioni di I/O, stringhe, operazioni su file. Ordinamento lessicografico di stringhe.
- Realizzazione di una libreria di funzioni ricorsive per la manipolazione di liste.

## MODALITÀ DIDATTICHE

Lezioni frontali e esercitazioni.

## MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta libera, esercizi numerici, sviluppo di piccoli progetti/esercizi, algoritmi e software.

## Architettura degli Elaboratori

### TITOLO INSEGNAMENTO IN INGLESE *Computer Architecture*

|                |                                                                       |
|----------------|-----------------------------------------------------------------------|
| SSD            | INFO-01/A                                                             |
| CFU            | 9                                                                     |
| Anno           | Primo                                                                 |
| Semestre       | Primo                                                                 |
| Propedeuticità | Nessuna                                                               |
| Docenti        | Canale 1: Luigi Sauro; Canale 2: Silvia Rossi; Canale 3: da definire. |

#### 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.

#### MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

#### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**.

Per la prova scritta, i quesiti sono a risposta multipla e/o a risposta libera.

## Analisi Matematica I

### TITOLO INSEGNAMENTO IN INGLESE *Calculus I*

|                |                                                                                            |
|----------------|--------------------------------------------------------------------------------------------|
| SSD            | MATH-03/A                                                                                  |
| CFU            | 9                                                                                          |
| Anno           | Primo                                                                                      |
| Semestre       | Primo                                                                                      |
| Propedeuticità | Nessuna                                                                                    |
| Docenti        | Canale 1: Roberto Alicandro; Canale 2: Daniele Castorina; Canale 3: Anna Maria Barbagallo. |

### 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

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.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta libera e/o esercizi numerici.

## Programmazione Object-Oriented

### TITOLO INSEGNAMENTO IN INGLESE *Object-Oriented Programming*

|                |                                                                                           |
|----------------|-------------------------------------------------------------------------------------------|
| SSD            | INFO-01/A                                                                                 |
| CFU            | 9                                                                                         |
| Anno           | Primo                                                                                     |
| Semestre       | Secondo                                                                                   |
| Propedeuticità | Nessuna                                                                                   |
| Docenti        | Canale 1: Sergio Di Martino; Canale 2: Porfirio Tramontana; Canale 3: Riccardo Caccavale. |

#### 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. Il corso fornisce anche linee guida e best practice di sviluppo, gestione ed organizzazione del codice sorgente per migliorarne la manutenibilità, la riutilizzabilità e l'efficienza.

#### 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.

#### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

#### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e realizzazione di un elaborato progettuale.**

Per la prova scritta, i quesiti sono a risposta libera, esercizi numerici.

## Basi di dati

### TITOLO INSEGNAMENTO IN INGLESE *Data Bases*

|                |                                                                            |
|----------------|----------------------------------------------------------------------------|
| SSD            | INFO-01/A                                                                  |
| CFU            | 9                                                                          |
| Anno           | Primo                                                                      |
| Semestre       | Secondo                                                                    |
| Propedeuticità | Nessuna                                                                    |
| Docenti        | Canale 1: Mara Sangiovanni; Canale 2: Silvio Barra; Canale 3: Andrea Calì. |

### 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

Introduzione alla Gestione dei Dati. Dati e sistemi informativi. Gestione con file vs gestione con DBMS. Architettura e funzionalità di un DBMS. Modellazione Concettuale. Modello Entità-Relazione (ER). Diagrammi UML per la rappresentazione dei dati. Progettazione concettuale. Ristrutturazione del modello concettuale. Algebra Relazionale. Operatori fondamentali e derivati. Esecuzione di interrogazioni tramite algebra relazionale. Confronto con SQL. Progettazione Logica e Fisica. Mapping concettuale-logico. Progettazione logica e normalizzazione. Progettazione fisica: indici, organizzazione dei file. SQL: Linguaggio per Basi di Dati. DDL (Data Definition Language). DML (Data Manipulation Language). SDL (Storage and access paths). Query con join, aggregazioni, sottoquery e viste. Integrità dei Dati. Vincoli di chiave, dominio, referenziali. Assertion e trigger. PL/SQL. Strutture di controllo, variabili e tipi. Procedure, funzioni e pacchetti. Gestione delle eccezioni. SQL Dinamico. Costruzione ed esecuzione dinamica di comandi SQL. Applicazioni tipiche in contesti multiutente o interattivi.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e realizzazione di un elaborato progettuale.**

Per la prova scritta, i quesiti sono a risposta libera, esercizi numerici.

## Algebra

### TITOLO INSEGNAMENTO IN INGLESE *Algebra*

|                |                                                                                         |
|----------------|-----------------------------------------------------------------------------------------|
| SSD            | MATH-02/A                                                                               |
| CFU            | 6                                                                                       |
| Anno           | Primo                                                                                   |
| Semestre       | Secondo                                                                                 |
| Propedeuticità | Nessuna                                                                                 |
| Docenti        | Canale 1: Maria Rosaria Celentani; Canale 2: Giovanni Cutolo; Canale 3: Mattia Brescia. |

### 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 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. Operazioni e strutture algebriche. Semigrupp, 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.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**.

La prova scritta prevede sia risposta multipla che risposta libera.

## Secondo Anno

### Analisi e progettazione di Algoritmi

TITOLO INSEGNAMENTO IN INGLESE *Algorithm Analysis and Design*

|                |                                                          |
|----------------|----------------------------------------------------------|
| SSD            | INFO-01/A                                                |
| CFU            | 6                                                        |
| Anno           | Secondo                                                  |
| Semestre       | Secondo                                                  |
| Propedeuticità | Analisi matematica I, Programmazione                     |
| Docenti        | Canale 1: Massimo Benerecetti; Canale 2: Fabio Mogavero. |

#### OBIETTIVI FORMATIVI

Il corso intende fornire un'introduzione alle tecniche di analisi del tempo di esecuzione degli algoritmi. Verranno presentate inoltre tecniche di progettazione di algoritmi, quali divide-et-impera, algoritmi greedy, programmazione dinamica, algoritmi approssimati con applicazioni alla soluzione di problemi di ottimizzazione. Infine, si forniranno cenni di complessità computazionale e trattabilità dei problemi.

#### CONTENUTI

Il corso fornisce gli strumenti fondamentali per l'analisi e la progettazione di algoritmi, con particolare attenzione alla loro correttezza e complessità sia temporale sia di memoria. Vengono introdotti i concetti di base della notazione asintotica e le tecniche per il calcolo del tempo di esecuzione di algoritmi iterativi e ricorsivi, inclusa la risoluzione di equazioni di ricorrenza. Particolare attenzione viene riservata all'analisi della complessità dei principali algoritmi di ordinamento (insertion sort, selection sort, merge sort, heap sort, quick sort). Si affronta il problema della correttezza degli algoritmi, con riferimento alle dimostrazioni per induzione e alla verifica della correttezza di algoritmi ricorsivi. Il corso esplora le principali tecniche di progettazione algoritmica: divide et impera, backtracking, programmazione dinamica e approcci greedy. Gli algoritmi di programmazione dinamica e greedy sono trattati attraverso applicazioni a problemi di ottimizzazione, con lo scopo di evidenziare criteri di scelta e limiti di ciascun approccio.

#### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

#### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**

In caso di prova scritta i quesiti sono: A risposta libera

## Analisi e progettazione di Strutture Dati

TITOLO INSEGNAMENTO IN INGLESE *Analysis and Design of Data Structures*

|                |                                                          |
|----------------|----------------------------------------------------------|
| SSD            | INFO-01/A                                                |
| CFU            | 9                                                        |
| Anno           | Secondo                                                  |
| Semestre       | Primo                                                    |
| Propedeuticità | Programmazione                                           |
| Docenti        | Canale 1: Massimo Benerecetti; Canale 2: Fabio Mogavero. |

### OBIETTIVI FORMATIVI

Il corso si propone di fornire le conoscenze di base per l'analisi e la progettazione di strutture dati efficienti. In particolare, verranno illustrate le tecniche di base per la valutazione dell'efficienza spaziale e temporale. Tali concetti verranno illustrati a livello teorico e metodologico e applicati, a titolo esemplificativo, all'analisi di strutture dati elementari (tra cui liste, alberi e grafi) e strutture dati avanzate (come, ad esempio, code a priorità, tabelle hash e alberi bilanciati). Ulteriore obiettivo è quello di familiarizzare lo studente con la progettazione e l'implementazione di strutture dati in un linguaggio di programmazione procedurale o a oggetti.

### CONTENUTI

Il corso affronta lo studio delle principali strutture dati, con attenzione sia agli aspetti teorici sia a quelli progettuali e implementativi. Si inizierà con l'analisi delle strutture dati elementari, quali array e liste, sia ordinati sia non ordinati, oltre a stack, code, heap, code a priorità e alberi binari, includendo le operazioni fondamentali e le modalità di visita. Verranno poi trattate strutture più complesse, come gli alberi binari di ricerca, sia nella versione base sia in quelle bilanciate (AVL e alberi Rosso-Neri), approfondendo gli algoritmi di inserimento, cancellazione e ricerca. Una parte del corso è dedicata alle tabelle ad accesso diretto, con particolare attenzione alle tabelle hash. Seguirà lo studio dei grafi, con un focus sulle principali tecniche di rappresentazione e sugli algoritmi di attraversamento (BFS e DFS). Le visite sui grafi verranno analizzate anche nelle loro applicazioni pratiche, tra cui la ricerca di cammini minimi in grafi non pesati, la verifica dell'aciclicità, gli ordinamenti topologici e l'individuazione delle componenti fortemente connesse. Durante il corso sarà posta particolare attenzione alla progettazione e implementazione di tipi di dati astratti e di librerie di strutture dati modulari, che favoriscano l'indipendenza tra interfaccia e rappresentazione concreta dei dati.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**

In caso di prova scritta i quesiti sono: A risposta libera

## Elementi di Informatica Teorica

TITOLO INSEGNAMENTO IN INGLESE *Introduction to Theoretical Computer Science*

|                |                                                         |
|----------------|---------------------------------------------------------|
| SSD            | INFO-01/A                                               |
| CFU            | 6                                                       |
| Anno           | Secondo                                                 |
| Semestre       | Primo                                                   |
| Propedeuticità | Nessuna                                                 |
| Docenti        | Canale 1: Alessandro De Luca; Canale 2: Aniello Murano. |

### 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 Chomsky. Automi a pila e 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.

### MODALITÀ DIDATTICHE

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

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta libera e/o esercizi numerici.

## Fisica Generale I

TITOLO INSEGNAMENTO IN INGLESE *General Physics I*

|                |                                                          |
|----------------|----------------------------------------------------------|
| SSD            | PHYS-01/A                                                |
| CFU            | 6                                                        |
| Anno           | Secondo                                                  |
| Semestre       | Primo                                                    |
| Propedeuticità | Nessuna                                                  |
| Docenti        | Canale 1: Goffredo Chirco; Canale 2: Giampiero Esposito. |

### OBIETTIVI FORMATIVI

Il contenuto del corso di Fisica è stato concepito con lo scopo di fornire allo studente del corso di laurea in Informatica strumenti di analisi e di sintesi basati sul metodo sperimentale e sui principi fondamentali delle scienze fisiche per poter rappresentare e modellare i fenomeni fisici col metodo scientifico. Le metodologie acquisite 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.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta multipla, a risposta libera, e/o esercizi numerici.

## Geometria

### TITOLO INSEGNAMENTO IN INGLESE *Geometry*

|                       |                                                        |
|-----------------------|--------------------------------------------------------|
| <b>SSD</b>            | MATH-02/B                                              |
| <b>CFU</b>            | 6                                                      |
| <b>Anno</b>           | Secondo                                                |
| <b>Semestre</b>       | Secondo                                                |
| <b>Propedeuticità</b> | Nessuna                                                |
| <b>Docenti</b>        | Canale 1: Francesca Cioffi; Canale 2: Marco Trombetti. |

### OBIETTIVI FORMATIVI

Si dovranno acquisire gli strumenti di base dell'algebra lineare e della geometria. L'obiettivo di questo insegnamento è, da un lato, quello di abituare lo studente ad affrontare problemi formali, utilizzando strumenti adeguati ed un linguaggio corretto, e dall'altro di risolvere problemi specifici di tipo algebrico e geometrico, con gli strumenti classici dell'algebra lineare.

### 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.

### MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**.

Per la prova scritta, i quesiti sono a risposta multipla, a risposta libera, e/o esercizi numerici.

## Linguaggi di Programmazione

### TITOLO INSEGNAMENTO IN INGLESE *Programming Languages*

|                       |                                                         |
|-----------------------|---------------------------------------------------------|
| <b>SSD</b>            | INFO-01/A                                               |
| <b>CFU</b>            | 6                                                       |
| <b>Anno</b>           | Secondo                                                 |
| <b>Semestre</b>       | Secondo                                                 |
| <b>Propedeuticità</b> | Programmazione, Programmazione Object-Oriented          |
| <b>Docenti</b>        | Canale 1: Piero Andrea Bonatti; Canale 2: Marco Faella. |

#### 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 e logici.

#### 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.

#### MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

#### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**.

Per la prova scritta, i quesiti sono a risposta multipla e/o a risposta libera.

## Metodi Statistici per l'informazione

### TITOLO INSEGNAMENTO IN INGLESE Statistical Methods for Computer Science

|                       |                                              |
|-----------------------|----------------------------------------------|
| <b>SSD</b>            | IINF-03/A                                    |
| <b>CFU</b>            | 6                                            |
| <b>Anno</b>           | Secondo                                      |
| <b>Semestre</b>       | Secondo                                      |
| <b>Propedeuticità</b> | Analisi Matematica I                         |
| <b>Docenti</b>        | Canale 1: Marco Lops; Canale 2: Mario Tanda. |

#### OBIETTIVI FORMATIVI

Obiettivo dell'insegnamento è fornire gli strumenti metodologici ed operativi per l'uso di metodi probabilistici e statistici nell'elaborazione dei dati e nel trattamento dell'informazione, e segnatamente con il concetto di variabili e vettori aleatori – discreti o continui – e con la loro caratterizzazione statistica.

#### CONTENUTI

Elementi di analisi combinatoria. Definizione di probabilità (frequentistica/assiomatica) e caratterizzazione di variabili discrete. Medie e momenti di variabili discrete. Teorema fondamentale della media. Estensione a variabili continue. Variabili aleatorie multiple (vettori aleatori) e loro caratterizzazione. Casi notevoli: vettori aleatori Gaussiani. Informazione e sua misura. Compressione senza perdite. Statistica inferenziale o descrittiva. Modelli dei dati e loro rilevanza. Regole di decisione. Stima Bayesiana e non. Stima a minimo errore quadratico medio (MMSE) e stima Least Mean Square (LMS): algoritmo del gradiente e sue varianti. Estensione ai processi aleatori (cenni).

#### MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

#### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**.

Per la prova scritta, i quesiti sono esercizi numerici.

## Sistemi Operativi

### TITOLO INSEGNAMENTO IN INGLESE *Operating System*

|                |                                                    |
|----------------|----------------------------------------------------|
| SSD            | INFO-01/A                                          |
| CFU            | 9                                                  |
| Anno           | Secondo                                            |
| Semestre       | Secondo                                            |
| Propedeuticità | Architettura degli Elaboratori                     |
| Docenti        | Canale 1: Alberto Finzi; Canale 2: Walter Balzano. |

### 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.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni. L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta multipla, a risposta libera, e/o esercizi numerici.

## Terzo Anno

### AI Technologies

#### TITOLO INSEGNAMENTO IN INGLESE *AI Technologies*

|                       |                                                 |
|-----------------------|-------------------------------------------------|
| <b>SSD</b>            | INFO-01/A                                       |
| <b>CFU</b>            | 6                                               |
| <b>Anno</b>           | Terzo                                           |
| <b>Semestre</b>       | Secondo                                         |
| <b>Propedeuticità</b> | Algebra                                         |
| <b>Docenti</b>        | Canale 1: Anna Corazza; Canale 2: Silvia Rossi. |

#### OBIETTIVI FORMATIVI

Il corso si propone di mettere lo studente nelle condizioni di scegliere e applicare correttamente la tecnica di Intelligenza Artificiale (IA) maggiormente adatta al problema affrontato. Per ottenere questo risultato il corso introdurrà una panoramica delle diverse tecniche, includendo tra l'altro apprendimento automatico, compreso il deep learning, approcci logici ed euristici. A tale panoramica verranno affiancate modalità di formalizzazione applicate a diversi problemi di esempio, nei campi più tipici dell'IA, quali l'elaborazione del linguaggio naturale o la visione computazionale o applicazioni biomediche, e relative implicazioni etiche, ma anche esempi di problemi più immediatamente applicativi. Enfasi verrà posta sullo studio delle caratteristiche intrinseche e delle modalità di valutazione dei diversi approcci considerati.

#### CONTENUTI

Introduzione. Tipi di problema da risolvere con esempi. Metodi formali e metodi empirici. Approcci simbolici, statistici e machine learning. Problemi e tipo di risultato che ci si può attendere coi diversi approcci. Studi empirici rispetto ad analisi formale: protocolli e conclusioni che si possono trarre.

Contesti applicativi in cui c'è l'esigenza di garantire la correttezza e la completezza dei risultati e cenni sulle metodologie adottate.

Elaborazione del linguaggio naturale: approcci a regole, modelli probabilistici, deep learning. Large Language Models. Loro applicazione a campi diversi (visione computazionale, robotica, planning, altro).

Implicazioni etiche e sociali, ivi compreso l'impatto energetico e ambientale.

Introduzione al linguaggio Python. Principali librerie per il machine learning.

Uno o due seminari su applicazioni specifiche.

#### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

#### MODALITÀ DI ESAME

Scritto e orale.

## Reti e Programmazione Distribuita

TITOLO INSEGNAMENTO IN INGLESE *Networks and Distributed Programming*

|                |                                                 |
|----------------|-------------------------------------------------|
| SSD            | INFO-01/A                                       |
| CFU            | 9                                               |
| Anno           | Terzo                                           |
| Semestre       | Primo                                           |
| Propedeuticità | Algebra, Sistemi Operativi                      |
| Docenti        | Canale 1: Alberto Finzi, Canale 2: Andrea Cali. |

### 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, la progettazione e lo sviluppo di sistemi e applicazioni distribuite. 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 TCP/IP. Il corso approfondirà le tecniche di programmazione di rete e distribuita, con particolare attenzione alla realizzazione di applicazioni distribuite e concorrenti, multi-processo e multi-thread.

### CONTENUTI

#### 1. Introduzione alle reti di calcolatori

Scopi e applicazioni delle reti di calcolatori; concetti fondamentali; topologie e categorie di rete; evoluzione storica; service providers; struttura a livelli; modello ISO/OSI e modello di riferimento; protocolli.

#### 2. Architettura a livelli TCP/IP

Livello Applicazione: Interfacciamento dei programmi utente tramite architetture client-server o peer-to-peer per navigazione web (HTTP), risoluzione dei nomi (DNS), trasferimento file (FTP, BitTorrent), accesso remoto (SSH) e gestione della posta elettronica (SMTP, POP3, IMAP).

Livello di Trasporto: Trasferimento end-to-end dei dati e regolazione dell'affidabilità tramite protocolli non orientati alla connessione (UDP) o orientati alla connessione (TCP), con meccanismi di controllo degli errori, handshake, timeout, flusso e congestione.

Livello di Rete: Instradamento globale e inoltro dei pacchetti tramite router, indirizzamento logico (IPv4, IPv6), configurazione automatica (DHCP), traduzione degli indirizzi (NAT) e algoritmi di routing (distance-vector, link-state).

Livello di Accesso alla Rete (Data Link e Fisico): Collegamento hardware delle interfacce tramite tecnologia Ethernet, risoluzione degli indirizzi MAC (ARP), verifica dell'integrità dei dati (parità, CRC) e gestione delle collisioni sul canale trasmissivo (ALOHA, CSMA/CD).

#### 3. Sicurezza sulle reti

Nozioni fondamentali; principali attacchi; introduzione alla crittografia (simmetrica e asimmetrica); integrità del messaggio; AP e TLS/SSL; firewall e sistemi di rilevamento delle intrusioni.

#### 4. Programmazione concorrente.

Processi e thread; modelli di concorrenza; meccanismi di comunicazione e sincronizzazione; coordinamento delle attività concorrenti.

#### 5. Programmazione di rete e sistemi distribuiti.

Programmazione di rete mediante socket e principali API; protocolli UDP e TCP; progettazione di server concorrenti;

sviluppo di applicazioni distribuite e meccanismi di comunicazione e scambio dati tra servizi.

#### **MODALITÀ DIDATTICHE**

Il corso prevede lezioni ed esercitazioni in aula svolte dal docente.

#### **MODALITÀ DI ESAME**

Verifica scritta e orale.

## Tecniche di Programmazione Avanzata

TITOLO INSEGNAMENTO IN INGLESE *Advanced Programming*

|                |                                                                      |
|----------------|----------------------------------------------------------------------|
| SSD            | INFO-01/A                                                            |
| CFU            | 6                                                                    |
| Anno           | Terzo                                                                |
| Semestre       | Primo                                                                |
| Propedeuticità | Linguaggi di Programmazione, Programmazione Object-Oriented, Algebra |
| Docenti        | Canale 1: Marco Faella; Canale 2: da definire.                       |

### OBIETTIVI FORMATIVI

Il corso intende esporre gli studenti ad una gamma di funzionalità avanzate dei moderni linguaggi di programmazione, tra cui il polimorfismo parametrico e i meccanismi di supporto alla concorrenza. 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. Inoltre, gli studenti sapranno utilizzare con consapevolezza gli strumenti semi-automatici di programmazione assistita da agenti artificiali.

### CONTENUTI

- Richiami di linguaggi Java e C++.
- Operazioni di base su oggetti: confronto ordinale e di uguaglianza.
- Aspetti funzionali dei linguaggi imperativi: classi e funzioni anonime, interfacce funzionali, riferimenti a metodi.
- Polimorfismo parametrico e sue implementazioni: i generics e i template.
- Programmare con collezioni: insiemi, sequenze, array associativi.
- Programmazione multi-thread: comunicazione e sincronizzazione tra thread, i modelli di memoria.
- Tecniche di programmazione assistita dall'intelligenza artificiale.

### MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**.

## Tecnologie Web

### TITOLO INSEGNAMENTO IN INGLESE *Web Technologies*

|                       |                                                                 |
|-----------------------|-----------------------------------------------------------------|
| <b>SSD</b>            | INFO-01/A                                                       |
| <b>CFU</b>            | 9                                                               |
| <b>Anno</b>           | Terzo                                                           |
| <b>Semestre</b>       | Secondo                                                         |
| <b>Propedeuticità</b> | Programmazione Object-Oriented, Algebra                         |
| <b>Docenti</b>        | Canale 1: Luigi Lucio Libero Starace; Canale 2: Bernardo Breve. |

#### OBIETTIVI FORMATIVI

L'obiettivo del corso è fornire una panoramica completa dei concetti fondamentali, dei protocolli di rete applicativi, delle tecnologie e degli strumenti allo stato dell'arte per la progettazione e la realizzazione di applicazioni web moderne. Gli obiettivi specifici includono:

- Comprensione dei principi fondamentali che regolano il World Wide Web, compresi i principali protocolli di rete a livello applicativo, la struttura dei dati e le dinamiche di navigazione.
- Saper progettare e implementare applicazioni web moderne, sicure e responsive, attraverso l'analisi critica delle esigenze del progetto e la selezione mirata delle tecnologie e degli strumenti più adatti.
- Sviluppare le competenze e le metodologie necessarie per rimanere aggiornati autonomamente in un settore in continua evoluzione, attraverso l'adozione di best practice per il lifelong learning.

#### CONTENUTI

- Protocolli web applicativi: HTTP/HTTPS, WebSocket, DNS, etc. Il World Wide Web: Documenti e Iper testi.
- HTML (living standard): Elementi fondamentali; URLs; Form. Strumenti per sviluppatori nei moderni Web Browser;
- CSS: Nozioni di base; Selettori; l'algoritmo Cascade; Ereditarietà; Layout (Flexbox, Grid); Media Query; Fondamenti di Responsive Design;
- JavaScript (ES6+): Nozioni di base del linguaggio; ECMAScript 6 e il JavaScript "moderno"; Variabili; Funzioni, Oggetti; Strutture dati; Classi; Gestione di errori; Moduli; JavaScript in un Web Browser; Manipolazione del DOM; Web Storage API; Asincronismo (Promise); Richieste di Rete (fetch API);
- Introduzione alla programmazione server-side: nozioni di base, esempi con CGI e PHP;
- JavaScript in un ambiente di esecuzione server: Node.js; npm; debugging; Introduzione al session tracking;
- Framework: Introduzione ai Framework; Principio dell'Inversion of Control; Il framework Express; Concetti di base in Express: Routes, Middleware; Templating (con Pug); Esempi pratici; Esempio di utilizzo di un ORM (Sequelize); Vantaggi e Svantaggi di un ORM.
- API REST: Introduzione alle API REST; Motivazioni; Autenticazione con token JWT; versionamento ed evoluzione di API REST.
- Introduzione al concetto di Content Management System (CMS); Esempio di utilizzo del CMS WordPress; CMS Headless; Esempio di utilizzo del CMS Headless Strapi; Cenni di base di GraphQL.
- TypeScript: Nozioni di base; Transpiling verso codice JavaScript; Downleveling; Dichiarazione di tipi; Inferenza automatica di tipi; Introduzione al concetto di Tipizzazione Strutturale; Cenni di tipi generici; Altri tipi: never, any, unknown, void; Esempi pratici;
- Front-end Tooling: Preprocessor e Postprocessor di CSS; Sass (variabili, mixin, nesting, control flow, strutture dati di base); Esempi con Sass; Framework CSS; Bundling; Tree-shaking; Minification; Esempi utilizzando il tool Vite;
- Single Page Applications (SPA): Concetti di base; Vantaggi e Svantaggi rispetto alle applicazioni web tradizionali; Sfide nell'implementazione di SPA (routing, gestione dello stato, ri-utilizzo dei componenti); Esempio di

implementazione di una SPA utilizzando soltanto JavaScript. Framework per realizzare SPA: Angular/React, esempi.

- Sicurezza per applicazioni web: principali vulnerabilità, conseguenze, e tecniche di mitigazione; meccanismi di protezione dei moderni browser web (CORS; CSP)

### **MODALITÀ DIDATTICHE**

Lezioni frontali ed esercitazioni.

### **MODALITÀ DI ESAME**

L'esame si articola in prova: Scritta e orale.

Per la prova scritta, i quesiti sono a risposta multipla, a risposta aperta, ed esercizi numerici. Progetto obbligatorio di gruppo.

## Ingegneria del Software

### TITOLO INSEGNAMENTO IN INGLESE *Software Engineering*

|                       |                                                             |
|-----------------------|-------------------------------------------------------------|
| <b>SSD</b>            | INFO-01/A                                                   |
| <b>CFU</b>            | 9                                                           |
| <b>Anno</b>           | Terzo                                                       |
| <b>Semestre</b>       | Primo                                                       |
| <b>Propedeuticità</b> | Programmazione Object-Oriented, Algebra                     |
| <b>Docenti</b>        | Canale 1: Sergio Di Martino; Canale 2: Porfirio Tramontana. |

### OBIETTIVI FORMATIVI

Comprendere l'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

- Il concetto di Ingegneria del Software.
- Il Ciclo di Vita del Software e le relative fasi.
- La Qualità del Software.
- Il Concetto di Prodotto Software
- Concetti di Qualità Interna ed Esterna del Software
- Gli standard di qualità ISO 9126 e ISO 25010.
- L'Ingegneria dei Requisiti.
- I Requisiti Funzionali, non Funzionali e di Dominio.
- La Requirement Elicitation.
- I Modelli di Dominio nell'Ingegneria dei Requisiti.
- L'Euristica Entity-Boundary-Control. Modellazione di dominio per applicazioni gestionali.
- Importanza del disaccoppiamento tra logica, presentazione e dati.
- Lo Unified Modeling Language.
- Il Diagramma dei Casi d'uso.
- Richiami su Class e Sequence Diagrams
- Gli Statecharts.
- Il Software Design.
- Concetti di Architetture Software.
- Principi di Software Design. Coesione, Accoppiamento
- Architetture Software. Architetture Chiuse e Aperte, Architetture a 2, 3 e n livelli.
- Il Concetto di Clean Code
- La Virtualizzazione.
- Il Cloud Computing. Tipologie di Cloud. Gli Amazon Web Services
- Accenni di Containerizzazione
- La Verification & Validation
- Definizioni
- Livelli di Testing
- Automated Unit Testing; il Framework xUnit; Lo Scaffolding
- Generazione di Casi di Test

- Testing Black-Box; Classi di Equivalenza, Strategie di testing combinatoriale
- Testing White Box. Il Control Flow Graph. Criteri di copertura WB.
- I Modelli di Processo.
- Il modello a Cascata.
- I modelli Agili e SCRUM

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e Orale**.

Per la prova scritta, i quesiti sono a risposta libera ed esercizi numerici. Progetto obbligatorio di gruppo.

## Esami a Scelta Libera

### Algorithm design

TITOLO INSEGNAMENTO IN INGLESE *Algorithm design*

|                |                                                                                 |
|----------------|---------------------------------------------------------------------------------|
| SSD            | INFO-01/A                                                                       |
| CFU            | 6                                                                               |
| Anno           | Secondo                                                                         |
| Semestre       | Secondo                                                                         |
| Propedeuticità | Analisi e Progettazione di Strutture Dati, Analisi e Progettazione di Algoritmi |
| Docenti        | Canale unico: Fabio Mogavero.                                                   |

### 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.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**.

Per la prova scritta, i quesiti sono a risposta libera.

## Calcolo Numerico

TITOLO INSEGNAMENTO IN INGLESE *Numerical Analysis*

|                |                              |
|----------------|------------------------------|
| SSD            | MATH-05/A                    |
| CFU            | 6                            |
| Anno           | Secondo/Terzo                |
| Semestre       | Primo                        |
| Propedeuticità | nessuna                      |
| Docenti        | Canale unico: Luisa D'Amore. |

### 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.

### MODALITÀ DIDATTICHE

Lezioni frontali ed attività di laboratorio.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta libera ed esercizi numerici.

## Diritto dell'Informatica

### TITOLO INSEGNAMENTO IN INGLESE *Legal Informatics*

|                |                            |
|----------------|----------------------------|
| SSD            | GIUR-17/A                  |
| CFU            | 6                          |
| Anno           | Secondo                    |
| Semestre       | Secondo                    |
| Propedeuticità | nessuna                    |
| Docenti        | Canale unico: da definire. |

### OBIETTIVI FORMATIVI

Obiettivo del corso è quello di fornire allo studente consapevolezza dell'esistenza di problemi giuridico/normativi legati alle nuove tecnologie, nonché strumenti conoscitivi per comprendere meglio il diritto d'autore, gestire progetti e attività professionali con un maggiore grado di autonomia, sviluppare e utilizzare tecnologie informatiche in modo conforme alla legge, ed avere consapevolezza degli aspetti giuridici in merito alla privacy in sistemi informatici.

### CONTENUTI

**Introduzione al corso:** l'obiettivo del corso è quello di indicare le principali modalità che danno rilevanza all'informatica nel settore giuridico.

Ci si soffermerà pertanto sui seguenti istituti:

**Copyright/diritto d'autore.** La disciplina del diritto d'autore. Software, banche di dati, opere multimediali, siti web. Software proprietario e open source. La titolarità del software creato e contratto di libera professione. I diritti musicali e cinematografici. Peer to peer. Digital Rights Management. Tutela dei domain names.

**Proprietà industriale.** Il codice della proprietà industriale. Marchi, nomi a dominio e brevetti. Tutela brevettuale del software. Le rivendicazioni. Le banche dati dei brevetti nazionali e internazionali.

**Contratti software.** Il contratto e gli elementi essenziali del contratto. Nullità e annullabilità. Contratti tipici e atipici. Principali tipologie di contratti utili per l'ingegnere. Cessione dei diritti di utilizzazione economica del software. Licenza d'uso di software, licenze open source. Creative commons. Contratti di sviluppo software.

**Privacy e sicurezza.** Privacy e protezione dei dati personali nella società dell'informazione. Principi generali. Informativa e consenso. Diritti dell'interessato. Dati sensibili. Misure di sicurezza minime e idonee. Privacy e big data. Profilazione degli utenti in rete e comunicazioni commerciali - Il Regolamento Generale europeo sulla protezione dei dati (GDPR 679/2016).

**Firme elettroniche/digitali e documenti informatici.** Le firme elettroniche: tipologie e disciplina giuridica. Il documento informatico: definizione, validità e rilevanza giuridica. La validità legale di e-mail, pagine web, registri informatici. La certificazione della firma digitale. Posta elettronica certificata. Conservazione sostitutiva dei documenti informatici - Codice dell'amministrazione digitale (D.Lgs. 82/2005).

**Commercio elettronico.** Obblighi e responsabilità dei prestatori di servizi on-line. Responsabilità degli intermediari. Comunicazioni commerciali. I contratti conclusi con strumenti informatici. La tutela dei consumatori in rete - Codice del consumo. Diritto di recesso e clausole vessatorie.

**Reati informatici.** Danneggiamento o interruzione di un sistema informatico. Interferenza illecita in comunicazioni informatiche o telematiche. Frode informatica.

#### **MODALITÀ DIDATTICHE**

La didattica sarà articolata mediante lezioni frontali, esercitazioni seminariali anche con strumentazioni multimediali.

#### **MODALITÀ DI ESAME**

L'esame si articola in prova: **Orale e discussione di elaborato progettuale.**

## Computer Forensics

### TITOLO INSEGNAMENTO IN INGLESE Computer Forensics

|                       |                            |
|-----------------------|----------------------------|
| <b>SSD</b>            | INFO-01/A                  |
| <b>CFU</b>            | 6                          |
| <b>Anno</b>           | Secondo/Terzo              |
| <b>Semestre</b>       | Secondo                    |
| <b>Propedeuticità</b> | Nessuna.                   |
| <b>Docenti</b>        | Canale unico: da definire. |

#### 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.

#### CONTENUTI

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.

#### MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

#### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta multipla ed a risposta libera.

## Economia e Organizzazione Aziendale

### TITOLO INSEGNAMENTO IN INGLESE *Economics*

|                       |               |
|-----------------------|---------------|
| <b>SSD</b>            | IEGE-01/A     |
| <b>CFU</b>            | 6             |
| <b>Anno</b>           | Secondo/Terzo |
| <b>Semestre</b>       | Secondo       |
| <b>Propedeuticità</b> | Nessuna       |
| <b>Docenti</b>        | Da definire   |

#### 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.

#### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

#### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

## Istituzioni di Matematica II

### TITOLO INSEGNAMENTO IN INGLESE *Calculus II*

|                |                                                    |
|----------------|----------------------------------------------------|
| SSD            | MATH-03/A                                          |
| CFU            | 6                                                  |
| Anno           | Secondo/Terzo                                      |
| Semestre       | Primo                                              |
| Propedeuticità | Analisi Matematica I                               |
| Docenti        | Mutuato dal Corso di Laurea in Ottica e Optometria |

#### 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.

#### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

#### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta libera ed esercizi numerici.

## Logic for computer science

TITOLO INSEGNAMENTO IN INGLESE *Logic for computer science*

|                |                                    |
|----------------|------------------------------------|
| SSD            | INFO-01/A                          |
| CFU            | 6                                  |
| Anno           | Secondo/Terzo                      |
| Semestre       | Secondo                            |
| Propedeuticità | Nessuna                            |
| Docenti        | Canale unico: Massimo Benerecetti. |

**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, problemi e ragionamenti nel linguaggio della logica classica, nonché di verificare la correttezza di ragionamenti informali.

**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.

**MODALITÀ DIDATTICHE**

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

**MODALITÀ DI ESAME**

L'esame si articola in prova: **Scritta e orale**.

Per la prova scritta, i quesiti sono a risposta libera ed esercizi numerici.

## Operating systems for mobile, cloud and IoT

TITOLO INSEGNAMENTO IN INGLESE Operating systems for mobile, cloud and IoT

|                |                             |
|----------------|-----------------------------|
| SSD            | INFO-01/A                   |
| CFU            | 6                           |
| Anno           | Secondo                     |
| Semestre       | Primo                       |
| Propedeuticità | Sistemi Operativi           |
| Docenti        | Canale unico: Barra Silvio. |

### 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, il Cloud e IoT (Internet of Things).

### 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 I, 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.

### MODALITÀ DIDATTICHE

Lezioni frontali.

### MODALITÀ DI ESAME

L'esame si articola in prova: **solo orale**.

## Parallel and Distributed Computing

### TITOLO INSEGNAMENTO IN INGLESE Parallel and Distributed Computing

|                |                              |
|----------------|------------------------------|
| SSD            | INFO-01/A                    |
| CFU            | 6                            |
| Anno           | Secondo/Terzo                |
| Semestre       | Primo                        |
| Propedeuticità | Nessuna                      |
| Docenti        | Canale unico: Marco Lapegna. |

### 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.

### MODALITÀ 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.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale.**

Per la prova scritta, i quesiti sono a risposta libera ed esercizi numerici.

## Operation Research

### TITOLO INSEGNAMENTO IN INGLESE *Operation Research*

|                       |                                           |
|-----------------------|-------------------------------------------|
| <b>SSD</b>            | MATH-06/A                                 |
| <b>CFU</b>            | 6                                         |
| <b>Anno</b>           | Terzo                                     |
| <b>Semestre</b>       | Primo                                     |
| <b>Propedeuticità</b> | Analisi e Progettazione di Strutture Dati |
| <b>Docenti</b>        | Canale unico: Paola Festa.                |

### 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

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.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **Scritta e orale**.

Per la prova scritta, i quesiti sono a risposta libera ed esercizi numerici.

## Scientific Computing

TITOLO INSEGNAMENTO IN INGLESE *Scientific Computing*

|                |                                 |
|----------------|---------------------------------|
| SSD            | MATH-05/A                       |
| CFU            | 6                               |
| Anno           | Secondo/Terzo                   |
| Semestre       | Primo                           |
| Propedeuticità | Nessuna                         |
| Docenti        | Canale Unico: Eleonora Messina. |

### 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.

### MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni in laboratorio

### MODALITÀ DI ESAME

L'esame si articola in prova: **solo orale**.

Sviluppo di progetti e prova al calcolatore

## Multimedia Information Systems

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

|                       |                               |
|-----------------------|-------------------------------|
| <b>SSD</b>            | INFO-01/A                     |
| <b>CFU</b>            | 6                             |
| <b>Anno</b>           | Secondo/Terzo                 |
| <b>Semestre</b>       | Secondo                       |
| <b>Propedeuticità</b> | Nessuna.                      |
| <b>Docenti</b>        | Canale Unico: Walter Balzano. |

### 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.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

### MODALITÀ DI ESAME

L'esame si articola in prova: **solo orale**

## Schede degli Insegnamenti – Manifesto ante 2024-2025

### Primo Anno

#### Algebra (DISATTIVATO DALL'A.A. 2024/2025)

#### TITOLO INSEGNAMENTO IN INGLESE Algebra

Docenti: Canale 1 Maria Rosaria Celentani, Canale 2 Giovanni Cutolo, Canale 3 Mattia Brescia

| 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. Semigrupp, 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.

#### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

#### MODALITÀ 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                                   |                     |  |                   |   |                   |  |

**Analisi Matematica I****TITOLO INSEGNAMENTO IN INGLESE** *Calculus I**Docenti: Canale 1 Francesco Oliva, Canale 2 Daniele Castorina, Canale 3 Anna Maria Barbagallo*

| SSD    | CFU | Anno di corso |    |     | Semestre |    | Lingua   |
|--------|-----|---------------|----|-----|----------|----|----------|
|        |     | I             | II | III | I        | II | Italiano |
| MAT/05 | 9   | X             |    |     | X        |    | X        |

**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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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                                   |                     |   |                   |   |                   |   |

## Architettura degli Elaboratori

TITOLO INSEGNAMENTO IN INGLESE *Computer Architecture*

Docenti: Canale 1 Silvia Rossi, Canale 2 Luigi Sauro, Canale 3 Roberto Prevete

| 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.

## MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

## MODALITÀ 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                                   |                     |   |                   |   |                   |  |

**Fisica Generale I****TITOLO INSEGNAMENTO IN INGLESE** *General Physics I**Docenti: Canale 1 Da definire, Canale 2 Goffredo Chirco, Canale 3 Giampiero Esposito*

| 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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                                   |                     |   |                   |   |                   |   |

## Geometria

TITOLO INSEGNAMENTO IN INGLESE *Geometry*

Docenti: Canale 1 Francesca Cioffi, Canale 2 Marco Trombetti, Canale 3 Da definire

| SSD    | CFU | Anno di corso |    |     | Semestre |    | Lingua   |
|--------|-----|---------------|----|-----|----------|----|----------|
|        |     | I             | II | III | I        | II | Italiano |
| MAT/03 | 6   | X             |    |     |          | X  | X        |

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.

## MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

## MODALITÀ 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                                   |                     |   |                   |   |                   |   |

**Laboratorio di Programmazione (DISATTIVATO DALL'A.A. 2024/2025)**TITOLO INSEGNAMENTO IN INGLESE *Computer Programming Laboratory*

Docenti: Canale 1 Laura Bozzelli; Canale 2 Antonio Origlia; Canale 3 Andrea Apicella

| 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Svolgimento di esercitazioni in laboratorio.

**MODALITÀ 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 |  |                   |  |                   |   |

## Programmazione (DISATTIVATO DALL'A.A. 2024/2025)

## TITOLO INSEGNAMENTO IN INGLESE Computer Programming

Docenti: Canale 1 Giuliano Laccetti; Canale 2 Daniel Riccio, Canale 3 Francesco Isgrò

| 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.

## MODALITÀ DIDATTICHE

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

## MODALITÀ 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                                   | Sviluppo piccoli progetti/esercizi, algoritmi e software |   |                   |   |                   |   |

## Secondo Anno

## Algoritmi e Strutture Dati I (DISATTIVATO DALL'A.A. 2025/2026)

TITOLO INSEGNAMENTO IN INGLESE *Algorithms and Data Structures I*

Docenti: Canale I Massimo Benerecetti, Canale 2 Fabio Mogavero

| 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.

## MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

## MODALITÀ 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 |  |
|-------|--|

**Basi di dati I (DISATTIVATO DALL'A.A. 2025/2026)**TITOLO INSEGNAMENTO IN INGLESE *Data Bases I*

Docenti: Canale 1 Mara Sangiovanni, Canale 2 Silvio Barra

| 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 enunzia, 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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                                   |                     |   |                   |   |                   |   |

**Elementi di Informatica Teorica**TITOLO INSEGNAMENTO IN INGLESE *Introduction to Theoretical Computer Science*

Docenti: Canale 1 Alessandro De Luca, Canale 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: 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 Chomsky. Automi a pila e 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.

**MODALITÀ DIDATTICHE**

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

**MODALITÀ 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                                   |                     |   |                   |   |                   |   |

**Laboratorio di Algoritmi e Strutture Dati (DISATTIVATO DALL'A.A. 2025/2026)**TITOLO INSEGNAMENTO IN INGLESE *Algorithm and Data Structures Laboratory*

Docenti: Canale 1 Fabio Mogavero; Canale 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 affrontata 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali ed esercitazioni in laboratorio.

**MODALITÀ 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 |  |
| Altro                                   | Sviluppo progetti   |   |                   |  |                   |  |

## Linguaggi di Programmazione I (DISATTIVATO DALL'A.A. 2025/2026)

TITOLO INSEGNAMENTO IN INGLESE *Programming Languages I*

Docenti: Canale 1 Piero Bonatti, Canale 2 Marco Faella

| 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.

## MODALITÀ DIDATTICHE

Lezioni frontali ed esercitazioni.

## MODALITÀ 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                                   |                     |   |                   |   |                   |  |

**Object Orientation (DISATTIVATO DALL'A.A. 2025/2026)**TITOLO INSEGNAMENTO IN INGLESE *Object Orientation*

Docenti: Canale 1. Sergio Di Martino, Canale 2 Porfirio Tramontana

| 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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            |  |                   |   |                   |   |

## Sistemi Operativi I (DISATTIVATO DALL'A.A. 2025/2026)

TITOLO INSEGNAMENTO IN INGLESE *Operating System I*

Docenti: Canale 1. Walter Balzano, Canale 2. Alberto Finzi

| 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.

### MODALITÀ DIDATTICHE

Lezioni frontali. Esercitazioni.

#### MODALITÀ 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                                   |                     |   |                   |   |                   |   |

**Terzo Anno****Laboratorio di Sistemi Operativi (DISATTIVATO DALL'A.A. 2026/2027)**TITOLO INSEGNAMENTO IN INGLESE *Operating Systems Laboratory*

Docenti: Canale unico, Alessandra Rossi

| 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali, esercitazioni.

**MODALITÀ 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   |  |                   |   |                   |  |

**Calcolo delle Probabilità e Statistica (DISATTIVATO DALL'A.A. 2025/2026)**TITOLO INSEGNAMENTO IN INGLESE *Probability Theory and Statistics*

Docenti: Canale unico. Marco Lops

| SSD    | CFU | Anno di corso |    |     | Semestre |    | Lingua   |
|--------|-----|---------------|----|-----|----------|----|----------|
|        |     | I             | II | III | I        | II | Italiano |
| MAT/06 | 9   |               |    | X   |          | X  | X        |

**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 genitrici normali. Stima puntuale. Stimatori e relative proprietà. Metodi di costruzione degli stimatori- Cenni alla stima intervallare e ai test d'ipotesi.

**MODALITÀ 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.

**MODALITÀ 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                                   | 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 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. |
|--|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Ingegneria del Software Modulo A (DISATTIVATO DALL'A.A. 2026/2027)**TITOLO INSEGNAMENTO IN INGLESE *Software Engineering (Mod. A)*

Docenti: Canale unico. 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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 |  |                   |   |                   |   |

**Ingegneria del Software Modulo B (DISATTIVATO DALL'A.A. 2026/2027)**TITOLO INSEGNAMENTO IN INGLESE *Software Engineering (Mod. B)*

Docenti: Canale unico. Luigi Libero Lucio Starace

| 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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 |  |                   |   |                   |   |

**Reti di Calcolatori I (DISATTIVATO DALL'A.A. 2026/2027)**TITOLO INSEGNAMENTO IN INGLESE *Computer Network I*

Docenti: Canale unico. Riccardo Caccavale

| 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

**MODALITÀ 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.

#### MODALITÀ 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 |   |                   |   |                   |  |

**Esami a Scelta Vincolata (Tab. A)****Linguaggi di Programmazione II (DISATTIVATO DALL'A.A. 2026/2027)**TITOLO INSEGNAMENTO IN INGLESE *Programming Languages II*

Docenti: Canale unico. 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali ed esercitazioni.

**MODALITÀ 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 |  |
|-------|--|

**Tecnologie Web (DISATTIVATO DALL'A.A. 2026/2027)**TITOLO INSEGNAMENTO IN INGLESE *Web Technologies*

Docenti: Canale unico. Luigi Libero Lucio Starace

| 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**

- Il World Wide Web: Documenti e Iper testi; il protocollo HTTP.
- HTML (living standard): Elementi fondamentali; URLs; Form. Strumenti per sviluppatori nei moderni Web Browser;
- CSS: Nozioni di base; Selettori; l'algoritmo Cascade; Ereditarietà; Layout (Flexbox, Grid); Media Query; Fondamenti di Responsive Design;
- JavaScript (ES6+): Nozioni di base del linguaggio; ECMAScript 6 e il JavaScript "moderno"; Variabili; Funzioni, Oggetti; Strutture dati; Classi; Gestione di errori; Moduli; JavaScript in un Web Browser; Manipolazione del DOM; Web Storage API; Asincronismo (Promise); Richieste di Rete (fetch API);
- Introduzione alla programmazione server-side: nozioni di base, esempi con CGI e PHP;
- JavaScript in un ambiente di esecuzione server: Node.js; npm; debugging; Introduzione al session tracking;
- Frameworks: Introduzione ai Frameworks; Principio dell'Inversion of Control; Il framework Express; Concetti di base in Express: Routes, Middleware; Templating (con Pug); Esempi pratici; Esempio di utilizzo di un ORM (Sequelize); Vantaggi e Svantaggi di un ORM.
- API REST: Introduzione alle API REST; Motivazioni; Autenticazione con token JWT.
- Introduzione al concetto di Content Management System (CMS); Esempio di utilizzo del CMS WordPress; CMS Headless; Esempio di utilizzo del CMS Headless Strapi; Cenni di base di GraphQL.
- TypeScript: Nozioni di base; Transpiling verso codice JavaScript; Downleveling; Dichiarazione di tipi; Inferenza automatica di tipi; Introduzione al concetto di Tipizzazione Strutturale; Cenni di tipi generici; Altri tipi: never, any, unknown, void; Esempi pratici;
- Front-end Tooling: Preprocessor e Postprocessor di CSS; Sass (variabili, mixin, nesting, control flow, strutture dati di base); Esempi con Sass; Framework CSS; Bundling; Tree-shaking; Minification; Esempi utilizzando il tool Vite;
- Single Page Applications (SPA): Concetti di base; Vantaggi e Svantaggi rispetto alle applicazioni web tradizionali; Sfide nell'implementazione di SPA (routing, gestione dello stato, ri-utilizzo dei componenti); Esempio di implementazione di una SPA utilizzando soltanto JavaScript.

- Angular: Il framework front-end Angular per l'implementazione di SPA; Nozioni di base; Strumenti per sviluppatori; Componenti; Routing; Forms; Services; Dependency Injection; Route Guards; HttpClient; Interceptors; Esempi di implementazione di SPA con Angular.
- Web Application Security: Nozioni di base; Sicurezza a livello di rete e di applicazione; Principali tipi di attacchi informatici ad applicazioni web: Cross-site Scripting (XSS), Cross-site Request Forgery (CSRF o XSRF), SQL Injection; Attacchi alla sessione (Session hijacking); Validazione inadeguata dei dati in input; Tecniche per mitigare le vulnerabilità; Esempi pratici di exploit di vulnerabilità su applicazioni web di esempio;
- Testing di Applicazioni Web: testing di unità; nozioni di base su testing end-to-end; fragilità; flakyness; Esempi con il tool Playwright

**MODALITÀ DIDATTICHE**

Lezioni frontali.

**MATERIALE DIDATTICO****MODALITÀ DI ESAME**

|                                                |                                                                      |          |                          |          |                          |  |
|------------------------------------------------|----------------------------------------------------------------------|----------|--------------------------|----------|--------------------------|--|
| <b>L'esame si articola in prova</b>            | <b>Scritta e orale</b>                                               | <b>X</b> | <b>Solo scritta</b>      |          | <b>Solo orale</b>        |  |
| <b>In caso di prova scritta i quesiti sono</b> | <b>A risposta multipla</b>                                           | <b>X</b> | <b>A risposta libera</b> | <b>X</b> | <b>Esercizi numerici</b> |  |
| <b>Altro</b>                                   | Progetto consistente nello sviluppo di una moderna applicazione web. |          |                          |          |                          |  |

**Esami a Scelta Libera (Tab. B)****Algorithm design**TITOLO INSEGNAMENTO IN INGLESE *Algorithm design*

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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni

**MODALITÀ 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                                   |                     |  |                   |   |                   |  |

**Calcolo Numerico**TITOLO INSEGNAMENTO IN INGLESE *Numerical Analysis*

Docenti: Canale unico Luisa D'Amore

| SSD    | CFU | Anno di corso |    |     | Semestre |    | Lingua   |
|--------|-----|---------------|----|-----|----------|----|----------|
|        |     | I             | II | III | I        | II | Italiano |
| MAT/08 | 6   |               | X  |     | X        |    | X        |

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.

**MODALITÀ DIDATTICHE**

Lezioni frontali ed attività di laboratorio.

**MODALITÀ 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                                   |                     |   |                   |   |                   |   |

## Diritto dell'Informatica

TITOLO INSEGNAMENTO IN INGLESE *Legal Informatics*

Docenti: Canale unico Lucio Franzese

| SSD    | CFU | Anno di corso |    |     | Semestre |    | Lingua   |
|--------|-----|---------------|----|-----|----------|----|----------|
|        |     | I             | II | III | I        | II | Italiano |
| IUS/20 | 6   |               | X  |     |          | X  | X        |

Insegnamenti propedeutici previsti: nessuno

## OBIETTIVI FORMATIVI

Obiettivo del corso è quello di fornire allo studente consapevolezza dell'esistenza di problemi giuridico/normativi legati alle nuove tecnologie, nonché strumenti conoscitivi per comprendere meglio il diritto d'autore, gestire progetti e attività professionali con un maggiore grado di autonomia, sviluppare e utilizzare tecnologie informatiche in modo conforme alla legge, ed avere consapevolezza degli aspetti giuridici in merito alla privacy in sistemi informatici.

## CONTENUTI

Da definire

## MODALITÀ DIDATTICHE

Lezioni frontali.

## MODALITÀ 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                                   |                     |   |                   |   |                   |  |

**Computer Forensics****TITOLO INSEGNAMENTO IN INGLESE Computer Forensics***Docenti: Canale unico. Lorenzo Laurato*

| 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali ed esercitazioni.

**MODALITÀ 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                                       |                     |   |                   |   |                   |  |

**Economia e Organizzazione Aziendale**TITOLO INSEGNAMENTO IN INGLESE *Economics*

Docenti: Da definire

| 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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                                   |                     |   |                   |   |                   |  |

## Istituzioni di Matematica II

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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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                                   |                     |   |                   |   |                   |   |

## Logic for computer science

TITOLO INSEGNAMENTO IN INGLESE *Logic for computer science*

Docenti: Canale unico. Massimo Benerecetti

| 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.

## MODALITÀ DIDATTICHE

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

## MODALITÀ 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                                   |                     |   |                   |   |                   |   |

## Operating systems for mobile, cloud and IoT

TITOLO INSEGNAMENTO IN INGLESE Operating systems for mobile, cloud and IoT

Docenti: Canale unico. Porfirio Tramontana

| 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 I, 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.

## MODALITÀ DIDATTICHE

Lezioni frontali.

## MODALITÀ 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                                   |                     |  |                   |  |                   |   |

## Parallel and Distributed Computing

TITOLO INSEGNAMENTO IN INGLESE *Parallel and Distributed Computing*

Docenti: Canale unico. 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.

## MODALITÀ 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.

## MODALITÀ 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                                   | Sviluppo di progetti durante il corso; Prova al calcolatore |   |                   |   |                   |   |

**Ricerca Operativa**TITOLO INSEGNAMENTO IN INGLESE *Operation Research*

Docenti: Canale unico. Paola Festa

| SSD    | CFU | Anno di corso |    |     | Semestre |    | Lingua   |
|--------|-----|---------------|----|-----|----------|----|----------|
|        |     | I             | II | III | I        | II | Italiano |
| MAT/09 | 6   |               | X  |     | X        |    | X        |

**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, organizza-zione, 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 Simplexso. 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 Simplexso. 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

#### MODALITÀ 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                                   |                     |   |                   |   |                   |   |

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

Docenti: Canale unico. Eleonora Messina

| SSD    | CFU | Anno di corso |    |     | Semestre |    | Lingua   |
|--------|-----|---------------|----|-----|----------|----|----------|
|        |     | I             | II | III | I        | II | Italiano |
| MAT/08 | 6   |               | X  |     | X        |    | X        |

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.

**MODALITÀ DIDATTICHE**

Lezioni frontali ed esercitazioni in laboratorio

**MODALITÀ 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 |  |                   |  |                   |   |

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

Docenti: Canale unico. 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.

**MODALITÀ DIDATTICHE**

Lezioni frontali. Esercitazioni.

**MODALITÀ 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                                   |                     |  |                   |  |                   |   |