

Regolamento DIDATTICO DEL CORSO DI LAUREA IN “INFORMATICA”

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

Indice

Art. 1 – Caratteristiche del progetto formativo	3
Art. 2 – Requisiti di ammissione al corso di studio	3
Art. 3 – Riconoscimento di attività formative	5
Art. 4 – Organizzazione del percorso formativo	7
Art. 5 – Piano di studio	8
Art. 6 – Opportunità di mobilità e altri servizi	9
Art. 7 – Conseguimento del titolo	10
Art. 8 – Sistema di assicurazione della qualità del CdS	10
Art. 9 – Norme finali e transitorie	11

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

Art. 1 – Caratteristiche del progetto formativo

- 1) Il presente Regolamento, che si applica alle coorti di studenti a decorrere dall'a.a. 2025/2026, disciplina gli aspetti organizzativi e didattici del corso di Studio in Informatica (di seguito anche CdS), attivato nella Classe L-31, Classe delle lauree in SCIENZE E TECNOLOGIE INFORMATICHE, di cui al DM 19/12/2023 n. 1648 ed è conforme a quanto previsto dall'Ordinamento didattico.
- 2) Le informazioni sul CdS sono presenti sul sito: <https://corsi.unitn.it/it/informatica-laurea>. Il Responsabile del CdS è indicato alla pagina web del CdS. L'Organismo di Gestione del CdS è il Consiglio di Dipartimento di Ingegneria e Scienza dell'Informazione, mentre le strutture di assicurazione della qualità sono specificate nel seguente Art. 8 del presente regolamento.
- 3) Gli obiettivi formativi specifici del CdS, i risultati di apprendimento attesi e agli sbocchi occupazionali e professionali, definiti nell'Ordinamento didattico, sono consultabili sulla pagina specifica del CdS all'interno di Course Catalogue, raggiungibile dal sito indicato al comma precedente, oppure consultando l'intero Course Catalogue all'indirizzo <https://unitn.coursecatalogue.cineca.it/>.
- 4) La struttura didattica di riferimento è il Dipartimento di Ingegneria e Scienza dell'Informazione. Le attività didattiche del CdS si svolgono (principalmente) presso la/le sede/i didattiche del Dipartimento.

Art. 2 – Requisiti di ammissione al corso di studio

- 1) I posti disponibili per l'iscrizione al primo anno sono stabiliti annualmente dagli Organi competenti e comunicati tempestivamente sul sito del CdS.
- 2) Ai sensi dell'Ordinamento, l'accesso al CdS è subordinato al possesso di un diploma di scuola secondaria di secondo grado conseguito in Italia o di altro titolo di studio conseguito all'estero, riconosciuto idoneo. In relazione agli obiettivi formativi specifici del Corso, è inoltre richiesto il possesso delle seguenti capacità e conoscenze:
 - Capacità
 - Capacità di comunicare efficacemente, in forma scritta e orale, in lingua italiana (per il percorso in italiano) o in lingua inglese (per il percorso in inglese).
 - Capacità di comprensione verbale: capacità di interpretare correttamente il significato di un testo o di una lezione, di effettuarne una sintesi per iscritto e di rispondere a quesiti basati sul suo contenuto.



REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

- Capacità logico-deduttive elementari (ad esempio, saper distinguere tra condizione necessaria e sufficiente, tra esempio e controesempio, tra necessità e possibilità).
- Capacità di astrazione: saper distinguere tra aspetti rilevanti ed irrilevanti in un problema.
- Capacità di individuare i dati di un problema e di utilizzarli per pervenire alla soluzione.
- Capacità di dedurre il comportamento di un sistema semplice partendo dalle leggi fondamentali e dalle caratteristiche dei suoi componenti.
- Capacità di collegare i risultati alle ipotesi che li determinano.
- Conoscenze linguistiche
 - conoscenza della lingua italiana e conoscenza della lingua inglese secondo quanto indicato nel bando di ammissione per i percorsi impartiti in lingua italiana;
 - conoscenza adeguata della lingua inglese (livello B2 o superiore) per i percorsi impartiti in lingua inglese.
- Conoscenze matematiche:
 - Aritmetica e algebra: proprietà e operazioni sui numeri (interi, razionali, reali); valore assoluto; potenze e radici; logaritmi ed esponenziali; calcolo letterale; polinomi (operazioni, decomposizione in fattori); equazioni e disequazioni algebriche di primo e secondo grado; sistemi di equazioni di primo grado.
 - Geometria: segmenti ed angoli (loro misura e proprietà); rette e piani; luoghi geometrici notevoli; proprietà delle principali figure geometriche piane; proprietà delle principali figure geometriche solide.
 - Geometria analitica e funzioni: coordinate cartesiane; concetto di funzione; equazioni di rette e di semplici luoghi geometrici; grafici e proprietà delle funzioni elementari.
 - Trigonometria: grafici e proprietà delle funzioni trigonometriche; principali formule trigonometriche (addizione, sottrazione, duplicazione, bisezione); relazioni fra elementi di un triangolo.
- Conoscenze scientifiche:
 - Conoscenza di base del metodo scientifico.
 - Conoscenza delle nozioni elementari sulle grandezze fisiche e sulla struttura della materia.
 - Cultura scientifica di base.

3) Modalità di verifica della preparazione dei candidati all'accesso al Corso di Studio:

- a) Per l'accesso al Corso di Studio, i candidati devono sostenere un test d'ingresso ed una prova di

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

conoscenza della lingua inglese, secondo il livello stabilito annualmente dal Dipartimento.

- b) Il superamento della prova di conoscenza della lingua inglese non è richiesto ai candidati in possesso di certificazione di livello corrispondente o superiore a quello stabilito dal Dipartimento.
- c) Possono immatricolarsi al Corso di Studio gli studenti che superano il test di ingresso conseguendo un punteggio almeno pari alla soglia minima stabilita annualmente dal Consiglio di Dipartimento. Il numero totale massimo di studenti ammessi viene deciso annualmente dal Consiglio di Dipartimento.
- d) Possono altresì immatricolarsi gli studenti che, nell'ultima prova di ingresso a calendario, pur non avendo superato il test, conseguono un punteggio almeno pari a una seconda soglia stabilita annualmente dal Consiglio di Dipartimento. Gli studenti immatricolati che non hanno superato il test di ingresso e che non superano la prima prova in itinere degli insegnamenti del primo anno di Matematica e/o di Fisica sono obbligati a frequentare corsi intensivi di recupero di Matematica e/o di Fisica, secondo modalità stabilite dal Dipartimento, fatto salvo il superamento delle relative prove d'esame prima dell'avvio dei corsi di recupero.
- e) Per gli studenti che non ottemperano a tali obblighi formativi aggiuntivi, nei termini previsti dal Dipartimento, il superamento della prova d'esame degli insegnamenti del primo anno di Matematica e/o di Fisica risulta vincolante come regola di precedenza rispetto a tutte le altre prove d'esame previste nel piano di studi.
- f) Gli studenti che non superano la prova di conoscenza della lingua inglese (livello A2 per i percorsi in italiano, livello B2 per i percorsi in inglese) sono tenuti a ripetere la prova nel corso del primo anno, secondo modalità stabilite dal Dipartimento. Il superamento della prova risulta vincolante come regola di precedenza rispetto alle prove di esame degli insegnamenti successivi al primo anno.
- g) Al fine di massimizzare la loro efficacia in modo tempestivo, le modalità sopra-descritte in questo Articolo possono subire variazioni che vengono decise dal Consiglio di Dipartimento.

Art. 3 – Riconoscimento di attività formative

- 1) A fronte della richiesta di riconoscimento di Crediti Formativi Universitari (CFU) acquisiti esternamente al CdS, viene sempre verificata la coerenza degli obiettivi formativi delle attività formative con gli obiettivi formativi specifici del CdS.
- 2) L'esito del riconoscimento in termini di CFU dipende in ogni caso anche dalle attività formative e relativi CFU che lo/la studente ha già acquisito e che sono utili ai fini del conseguimento del titolo rilasciato al

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

termine del CdS.

- 3) Nei casi di trasferimento da altro CdS trova inoltre applicazione quanto previsto dal DM 1648/2023 all'articolo 3 commi 10 e 11. Qualora il CdS dovesse prevedere la programmazione degli accessi, il numero di posizioni disponibili per gli anni successivi al primo è definito annualmente dalla differenza tra il numero programmato e gli studenti effettivamente iscritti. Nel caso di posti disponibili, l'ammissione da trasferimento da altro CdS è disciplinata mediante appositi avvisi.
- 4) Il passaggio dal percorso **Scienze e Tecnologie Informatiche** (in lingua italiana) al percorso **Computer Sciences and Technologies** (in lingua inglese), o viceversa, è soggetto ai seguenti vincoli:
 - a) è possibile solo a partire dall'inizio del secondo anno in poi;
 - b) è condizionato alla disponibilità di posti nel percorso entrante. In caso di numero di richieste superiore al numero di posti, le richieste vengono selezionate in funzione del numero di crediti acquisiti e della relativa media.
- 5) Ai sensi del DM 04/08/2024 n. 931 possono essere riconosciuti fino a 48 CFU nei seguenti casi:
 - a) conoscenze e abilità professionali, certificate ai sensi della normativa vigente in materia, nonché altre conoscenze e abilità maturate in attività formative di livello post-secondario;
 - b) attività formative svolte nei cicli di studio presso gli istituti di formazione della pubblica amministrazione, nonché altre conoscenze e abilità maturate in attività formative di livello post-secondario, alla cui progettazione e realizzazione l'Università abbia concorso;
 - c) conseguimento da parte dello Studente di medaglia olimpica o paralimpica ovvero del titolo di campione mondiale assoluto, campione europeo assoluto o campione italiano assoluto nelle discipline riconosciute dal Comitato olimpico nazionale italiano o dal Comitato italiano paralimpico.

Le domande di riconoscimento di tali CFU saranno valutate in modo insindacabile da un'apposita commissione nominata dall'Organismo di Gestione tenendo conto della stretta coerenza con gli obiettivi formativi e i risultati di apprendimento attesi del CdS. Lo studente dovrà indicare gli insegnamenti del CdS per i quali intende chiedere il riconoscimento dei CFU, e dovrà presentare una documentazione completa sul conseguimento degli obiettivi e sui risultati dell'apprendimento di cui sopra, inclusa un'indicazione dettagliata del numero di ore svolte nell'ambito delle attività che si chiede vengano riconosciute, e delle modalità di valutazione delle stesse, che in ogni caso dovranno essere coerenti con le modalità di valutazione degli insegnamenti offerti nel CdS.

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

Art. 4 – Organizzazione del percorso formativo

- 1) Le attività formative complete dei relativi obiettivi formativi sono elencate nell'allegato 1.
- 2) La struttura del corso di studio, la sua articolazione in percorsi, nonché i vincoli per la costruzione del piano di studi sono riportati nell'allegato 2 (offerta didattica programmata).
- 3) L'offerta didattica erogata in ogni anno accademico è pubblicata nel Manifesto degli studi.
- 4) Le attività didattiche possono comprendere lezioni frontali, esercitazioni in aula e in campo, attività di laboratorio, attività di tutorato, seminari e tirocini formativi. Le modalità di svolgimento degli insegnamenti e delle altre attività formative e le modalità di verifica dell'apprendimento, vengono indicate dai docenti responsabili prima dell'inizio di ogni anno accademico tramite la pubblicazione del syllabus.
- 5) Il CdS inoltre promuove l'acquisizione di conoscenze e competenze anche tramite open badge e microcredenziali rilasciati da Istituzioni soggette a un processo di accreditamento, in particolare per le attività rientranti nelle "altre attività", nelle attività "ad autonoma scelta", nelle attività affini e integrative. L'eventuale riconoscimento di open badge e microcredenziali è sempre subordinato alla verifica della loro coerenza rispetto agli obiettivi formativi specifici del CdS.
- 6) L'impegno richiesto allo studente per ogni attività formativa è misurato in Crediti Formativi Universitari (CFU). Un credito corrisponde a circa 25 ore di impegno complessivo per lo studente, comprese quelle dedicate allo studio individuale. Ogni CFU prevede in particolare:
 - a) Per le attività che consistono in corsi di insegnamento, ogni CFU comporta un numero medio di ore di lezione pari a 8 ore per credito, variabile fra 6 e 10 ore di lezione per credito tenendo conto della specificità del settore scientifico disciplinare e dell'eventuale presenza di attività progettuali a carico dello studente.
 - b) Per le attività che consistono in corsi di laboratori, ogni CFU comporta da 3 a 4 ore di lezione o esercitazione in aula e da 4 a 6 ore di attività di laboratorio che hanno carattere di sperimentazione guidata e mirano a sviluppare le capacità dello studente di applicare sperimentalmente le conoscenze sviluppate nel corso di studio.
 - c) Per tutti gli altri corsi, ogni credito comporta da 5 ad 8 ore di lezione o esercitazione in aula e, laddove appropriato, fino a 4 ore dedicate alle esercitazioni attive in aula o laboratorio da parte degli studenti che hanno carattere di studio guidato e mirano a sviluppare le capacità dello studente nel risolvere problemi ed esercizi.
 - d) per il tirocinio 25 ore di impegno per ogni CFU.

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

- e) Per i corsi di lingua straniera effettuati dal CLA (Centro Linguistico di Ateneo) può essere previsto un diverso rapporto ore/CFU.
 - f) Il tempo riservato allo studio personale e ad altre attività formative di tipo individuale è pari almeno al 60% dell'impegno orario complessivo.
- 7) Per ciascun esame o verifica del profitto è individuato un/a docente responsabile della procedura di valutazione, il/la quale ne garantisce il corretto svolgimento. Il/la docente responsabile della procedura di valutazione, che di norma è il/la titolare dell'attività formativa, garantisce il corretto svolgimento della procedura e ne registra tempestivamente il risultato nel sistema informatico dell'Ateneo. Il/la docente responsabile può essere coadiuvato/a da altre persone scelte nell'ambito di un insieme di docenti ed altri/e esperti/e individuati/e quali componenti della Commissione d'esame. Nel caso di attività formative articolate in più unità didattiche, il cui svolgimento risulti affidato a più docenti, la verifica finale del profitto è in ogni caso unitaria e collegiale.
- 8) La verifica dell'apprendimento può svolgersi in forma di esame orale e/o scritto. Tutte le prove orali sono pubbliche. Qualora siano previste prove scritte, la candidata/il candidato ha il diritto di prendere visione dei propri elaborati dopo la valutazione degli stessi. Le modalità di svolgimento delle verifiche sono riportate nel Syllabus di ciascun insegnamento. La valutazione è espressa in trentesimi con l'eventuale aggiunta della lode o, ove previsto, con due soli gradi ("approvato" o "non approvato").
- 9) La durata normale del CdS è di 3 anni e per conseguire il titolo finale si deve avere acquisito 180 CFU. Lo/la studente che abbia ottenuto tutti i CFU previsti prima della scadenza della durata normale del CdS, nel rispetto del presente Regolamento e più in generale delle norme e regolamenti di riferimento, può comunque conseguire il titolo di studio.

Art. 5 – Piano di studio

- 1) Ogni studente deve presentare il proprio piano di studi secondo le modalità stabilite annualmente. I piani di studi conformi all'offerta programmata del CdS/curriculum cui è iscritto lo studente sono approvati automaticamente.
- 2) Lo/la studente dovrà individuare anche gli insegnamenti a "autonoma/libera scelta" per un totale di 12 CFU, a completamento delle attività formative previste dal CdS. Tali insegnamenti possono essere selezionati tra gli insegnamenti elencati nel Manifesto degli studi del CdS, tra quelli offerti dal Dipartimento o anche tra quelli offerti da altri Dipartimenti purché coerenti con il percorso culturale dello studente. Tutti

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

gli insegnamenti presenti annualmente nel Manifesto sono approvati automaticamente. Tutti gli insegnamenti del Dipartimento offerti alla laurea triennale che non siano sostanzialmente equivalenti ad esami già presenti nel piano di studi dello studente sono approvati previa verifica da parte dell'Organismo di Gestione del CdS. L'Organismo di Gestione del CdS può predisporre tabelle di insegnamenti la cui inclusione o riconoscimento sono automatici. L'inclusione di un insegnamento di livello diverso dalla Laurea triennale deve essere opportunamente motivata ed è soggetta a valutazione. La richiesta di inserimento, tra i corsi a scelta, di insegnamenti offerti da CdS di altri Dipartimenti deve essere corredata di opportune motivazioni. Nei casi in cui nella compilazione online del piano di studi non sia possibile per lo/la studente selezionare insegnamenti che intenderebbe inserire nei CFU a libera scelta, è richiesta la presentazione, con altre modalità, di un'istanza corredata dalle opportune motivazioni. L'Organismo di Gestione del CdS, anche avvalendosi di figure appositamente delegate, verifica la coerenza delle proposte rispetto agli obiettivi formativi del CdS e ha la facoltà di richiedere allo/a studente le necessarie modifiche. Non si ammette la selezione di corsi che siano già stati sostenuti in una precedente carriera.

- 3) Lo/la studente può inoltre, ai sensi della normativa vigente, proporre un piano di studi individuale, motivando adeguatamente la richiesta finalizzata a sostituire nel proprio piano di studi attività formative previste nell'offerta programmata della coorte cui appartiene. In ogni caso il piano di studio individuale, che deve rispettare l'ordinamento didattico del CdS dell'anno di immatricolazione, viene accettato o respinto con parere motivato dell'Organismo di gestione del CdS o dai suoi delegati.
- 4) Sono definiti annualmente nel Manifesto degli studi eventuali obblighi di frequenza associati alle attività formative. In questi casi il/la docente responsabile dell'attività formativa specifica nel syllabus le modalità di verifica della frequenza.

Art. 6 – Opportunità di mobilità e altri servizi

- 1) Il CdS incoraggia la mobilità nazionale e internazionale degli/delle studenti, considerandola un mezzo di scambio culturale e di integrazione per la formazione personale e professionale ai fini del conseguimento del titolo di studio. In particolare, riconosce i periodi di studio svolti presso istituzioni universitarie italiane e straniere. Questi periodi di studio sono considerati uno strumento di formazione analogo a quello offerto dal CdS, a parità di impegno dello/a studente e di coerenza dei contenuti con il percorso formativo.
- 2) Il Learning Agreement è lo strumento che definisce il progetto delle attività formative che lo/la studente seguirà presso l'altra istituzione universitaria e che sostituiranno alcune delle attività previste dal piano di

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

studi.

- 3) Accanto alle attività di orientamento e tutorato svolte dai docenti nell'ambito dei propri compiti istituzionali, il CdS promuove il servizio di tutorato sia nella forma di "tutorato alla pari" sia con assegni di tutorato destinati a specifiche figure di tutor disciplinari.
- 4) Le opportunità di mobilità internazionale offerte agli studenti e i requisiti di partecipazione richiesti sono indicati nei siti web del Dipartimento e dell'Ateneo.
- 5) Per gli/le studenti con disabilità, DSA o bisogni educativi speciali è attivo il servizio di tutorato specializzato coordinato dal Servizio inclusione studente di Ateneo che, anche grazie al supporto di studenti senior e in collaborazione con il/la docente delegato/a per la disabilità/inclusione del Dipartimento, garantisce agli/alle studenti la più ampia integrazione nell'ambiente di studio.
- 6) Gli/le studenti possono avvalersi del servizio di consulenza psicologica di Ateneo, che rappresenta uno spazio di ascolto e sostegno durante tutto il percorso universitario allo scopo di migliorare l'avanzamento nel percorso formativo e la qualità della vita universitaria.
- 7) Gli studenti che hanno necessità di assistenza possono anche fare diretto riferimento al Delegato per la disabilità in Dipartimento.

Art. 7 – Conseguimento del titolo

- 1) Lo studente può sostenere la prova finale dopo aver completato tutte le altre attività formative previste dal suo piano di studio. La prova finale è volta a valutare la maturità scientifica raggiunta dallo studente, l'autonomia di giudizio e la padronanza degli argomenti, la capacità di operare in modo autonomo e l'abilità di comunicazione. La discussione è rivolta anche a valutare la preparazione generale dello studente in relazione ai contenuti formativi appresi nel CdS.
- 2) L'eventuale elaborato oggetto della prova finale può essere redatto, anche solo parzialmente, nell'ambito di un'attività di stage, di tirocinio o dell'eventuale percorso doppio titolo.
- 3) Le procedure relative all'ammissione alla prova finale, al suo svolgimento, alla costituzione delle commissioni, nonché al conferimento del titolo sono disciplinate dal Regolamento del Dipartimento in materia di prova finale e conseguimento del titolo delle lauree.

Art. 8 – Sistema di assicurazione della qualità del CdS

- 1) Il CdS adotta un Sistema di Assicurazione della Qualità (AQ) in conformità con il Sistema di AQ dell'Ateneo, che si basa su una costante interazione con le organizzazioni rappresentative della

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

produzione di beni e servizi e che coinvolge tutti gli attori interessati (docenti, studenti, personale tecnico-amministrativo).

- 2) L'organo deliberante di gestione del CdS è il Consiglio di Dipartimento. Dal punto di vista operativo, il CdS è gestito di concerto con gli altri corsi di laurea e laurea magistrale tramite il Tavolo della Didattica, organo composto dal Delegato alla Didattica (che presiede) e dai Responsabili dei CdS del Dipartimento, coadiuvati dal Responsabile di Staff tecnico-amministrativo e dal Personale Amministrativo di supporto alla didattica. Il Tavolo della Didattica ha il compito principale di pianificare, gestire e monitorare tutte le attività legate alla didattica, oltre ad istruire tutti gli argomenti successivamente sottoposti a discussione ed approvazione al Consiglio di Dipartimento. Esso si riunisce di norma con frequenza settimanale. Ove si rende utile o necessario, alle riunioni vengono invitati i Rappresentanti degli Studenti, i Delegati del DISI, o altri Uffici e docenti interessati.
- 3) All'interno del corso di studio è operativo un gruppo di riesame (GdR) che svolge un costante monitoraggio delle iniziative realizzate e dei risultati prodotti, anche mediante la predisposizione della Scheda di monitoraggio annuale (SMA) e la redazione del Rapporto di riesame ciclico (RRC) a cadenza periodica, o quando ritenuto necessario dall'organismo di gestione del CdS o da altri attori del Sistema di AQ dell'Ateneo, nonché l'analisi degli esiti delle opinioni degli studenti sulla didattica.
- 4) Il GdR è costituito dal Coordinatore/trice del CdS da almeno un altro docente che abbia un incarico didattico all'interno del corso di studio e da almeno uno studente iscritto al CdS.
- 5) In attuazione del Regolamento del Dipartimento, il corso è rappresentato all'interno della Commissione paritetica docenti-studenti (CPDS):
 - a) direttamente, attraverso i/le docenti e gli/le studenti del corso;
 - b) o indirettamente, mediante confronti sistematici attivati dalla CPDS con il GdR e/o con docenti e studenti referenti del CdS.

Art. 9 – Norme finali e transitorie

- 1) Le disposizioni del presente Regolamento si applicano alle nuove carriere attivate nell'a.a. 2025-26 e seguenti, fatta salva l'emanazione di un nuovo Regolamento nel quale sarà indicato il relativo a.a di decorrenza.
- 2) Eventuali problematiche interpretative o applicative derivanti dalla successione dei Regolamenti Didattici nel tempo o eventuali deroghe o estensioni nell'applicazione del Regolamento verranno gestite dal

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

Direttore del Dipartimento o da un suo Delegato.

- 3) Per quanto non espressamente qui disciplinato si rinvia al Regolamento didattico di Ateneo, al Regolamento di Dipartimento, al Regolamento per le prove finali di Dipartimento e alla normativa vigente in materia.

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA
ALLEGATO 1: Tabella 1 – OBIETTIVI FORMATIVI DELLE ATTIVITÀ FORMATIVE PREVISTE DAL PERCORSO

*CORSO DI LAUREA IN INFORMATICA : OBIETTIVI DELLE ATTIVITÀ PREVISTE A PARTIRE
DALLA COORTE 2025-26*

NOME INSEGNAMENTO	OBIETTIVI FORMATIVI
Advanced Programming	<p>L'insegnamento si propone di fornire agli studenti:</p> <ul style="list-style-type: none"> - l'abilità di scrivere e comprendere programmi scritti in Rust; - la conoscenza della semantica di Rust e della sua gestione di memoria e puntatori; - l'abilità di sviluppare un progetto in gruppo, in Rust, utilizzando librerie sviluppate da altri studenti. <p>Inoltre il corso si propone di fornire agli studenti:</p> <ul style="list-style-type: none"> - la comprensione del funzionamento di un W3C-style working group; - l'abilità di organizzare e gestire un W3C-style working group. <p>Pertanto l'insegnamento alterna lezioni frontali per l'apprendimento di Rust (durante le quali saranno usati quiz ed altri metodi di engagement) a lezioni di working group. Durante il working group la classe si organizza in un W3C-style working group e giunge alla stesura delle specifiche del progetto partendo da specifiche errate fornite dal docente.</p> <p>Al termine dell'insegnamento lo studente sarà in grado di:</p> <ul style="list-style-type: none"> - sviluppare programmi in Rust e testarne la correttezza; - gestire la produzione, la pubblicazione e l'integrazione di programmi in Rust tramite un repository; - identificare i ruoli di un W3C-style working group; - data una problematica, organizzare un W3C-style working group che si organizzi e risolva tale problematica.
Algorithms and Data Structures	The course aims to provide students with both theoretical and practical knowledge to design and implement efficient algorithmic solutions to solve a wide range of

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	computational problems.
Algoritmi e strutture dati	<p>L'obiettivo del corso è fornire agli studenti gli strumenti teorici e pratici per affrontare la progettazione di soluzioni algoritmiche corrette ed efficienti per una vasta gamma di problemi computazionali. Al termine del corso, gli studenti dovranno essere in grado di:</p> <p>A) Elencare le caratteristiche delle principali strutture dati e comprendere il loro contributo nella soluzione di problemi computazionali.</p> <p>B) Descrivere in modo formale una selezione rappresentativa di problemi computazionali e le relative soluzioni, discutendo la loro correttezza.</p> <p>C) Conoscere le principali tecniche algoritmiche di risoluzione e selezionare quelle più adatte per risolvere un problema specifico.</p> <p>D) Analizzare la complessità computazionale degli algoritmi considerati, quantificando le risorse necessarie per la loro esecuzione.</p> <p>E) Confrontare diverse soluzioni algoritmiche, selezionando quella più adatta al problema e all'input specifico.</p> <p>Forte di queste conoscenze e competenze generali sulla letteratura degli algoritmi "classici", gli studenti saranno in grado di affrontare una vasta gamma di problemi computazionali, ovvero:</p> <p>F) Comprendere con precisione il problema da risolvere, definendo la relazione matematica tra l'input del problema e l'output desiderato.</p> <p>G) Progettare una soluzione algoritmica per il problema, adattando una soluzione esistente o creandone una nuova.</p> <p>H) Valutare la correttezza e la complessità computazionale della soluzione proposta.</p> <p>I) Implementare la soluzione scelta utilizzando un linguaggio di programmazione imperativo.</p>
Analisi matematica 1	<p>1. Conoscenza e capacità di comprensione</p> <p>Conoscenza approfondita degli argomenti di base del calcolo di funzioni di una variabile come specificato dal programma del corso. Capacità di analizzare e di comprenderne le principali caratteristiche delle funzioni a valori reali differenziabili e</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>integrabili. Capacità di applicare i metodi a problemi concreti e ad altri problemi scientifici o della industria.</p> <p>2. Capacità di applicare conoscenza e comprensione</p> <p>Capacità di ragionamento induttivo e deduttivo nell'affrontare problemi della natura dell'analisi delle funzioni. Capacità di impostare un problema e di risolverlo utilizzando opportuni strumenti dell'analisi.</p> <p>3. Autonomia di giudizio</p> <p>Capacità di sviluppare argomentazioni logiche e produrre dimostrazioni corrette. Capacità di individuare i metodi più appropriati per analizzare e interpretare problemi.</p> <p>4. Abilità comunicative</p> <p>Capacità di esporre argomenti dell'analisi matematica.</p> <p>5. Capacità di apprendimento</p> <p>Capacità di acquisire e gestire nuove informazioni a partire dagli enunciati e dimostrazioni sviluppate a lezioni.</p>
Analisi matematica 2	<p>1. Conoscenza e capacità di comprensione</p> <p>Lo studente deve raggiungere una conoscenza approfondita degli argomenti di base del calcolo differenziale e integrale per le funzioni di due o più variabili reali. Lo studente deve acquisire la capacità di utilizzare un linguaggio matematico corretto e di saper fare brevi dimostrazioni.</p> <p>2. Capacità di applicare conoscenza e comprensione</p> <p>Lo studente deve acquisire capacità di ragionamento induttivo e deduttivo nell'affrontare problemi e la capacità di schematizzare un fenomeno fisico o geometrico in termini rigorosi.</p> <p>3. Autonomia di giudizio</p> <p>Lo studente deve essere in grado di sviluppare semplici argomentazioni logiche, di riconoscere la correttezza di semplici dimostrazioni e di produrre semplici dimostrazioni corrette.</p> <p>Lo studente deve essere in grado di individuare i metodi più appropriati per analizzare ed affrontare problemi risolvibili con le tecniche acquisite.</p>



REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>4. Abilità comunicative</p> <p>Lo studente deve essere in grado di esporre argomenti di base legati all'analisi matematica in un linguaggio corretto.</p> <p>5. Capacità di apprendimento</p> <p>Lo studente deve essere in grado di acquisire autonomamente e gestire nuove informazioni inerenti a tecniche e problemi legati all'analisi matematica.</p>
Basi di dati	<p>I dati ormai pervadono il nostro mondo. Quasi tutte le attività umane, ad oggi, producono dati o necessitano di accedere ai dati. In tal senso, sono quindi necessarie soluzioni che permettano la memorizzazione, il mantenimento, la ricerca, e l'estrazione di grandi quantità di dati, in modo efficace ed efficiente. I sistemi di gestione dati si occupano proprio di questo.</p> <p>L'obiettivo di questo corso è di introdurre agli studenti i concetti fondamentali della gestione dei dati, di impartire i buoni principi di progettazione delle basi di dati, di permettere agli studenti di acquisire le competenze di costruire interrogazioni complesse, e permettere loro di identificare e risolvere problemi relativi alle performance in contesti reali.</p> <p>Il corso si focalizzerà sulle basi di dati relazionali, in virtù del fatto che il modello relazionale costituisce, attualmente, le fondamenta della maggior parte dei sistemi di gestione dati utilizzati in ambito industriale. Concepito ormai più di 30 anni fa, il modello relazionale è il modello di dati più studiato, e vanta il maggior numero di strumenti commerciali attualmente attivi sul mercato.</p>
Calcolatori	<p>Il corso si propone di fornire le nozioni di base relative all'organizzazione e all'architettura delle macchine da calcolo, ovvero sia i comuni computer "general-purpose", sia le architetture specializzate progettate per effettuare compiti specifici. Il corso fornisce inoltre informazioni di base sul linguaggio assembly (in particolare utilizzando come riferimento RISC-V e accennando alle architetture Intel e ARM), sull'interfaccia HW/SW, e sul funzionamento di basso livello di un calcolatore. Il corso fornisce infine le basi necessarie per seguire eventuali corsi più avanzati dedicati al progetto di sistemi di elaborazione da un punto di vista dell'hardware.</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>Il corso non richiede alcuna nozione propedeutica. L'impostazione è pensata principalmente per studenti/esse con un curriculum dedicato al software e alla sistemistica, e che quindi seguono, nel corso triennale di laurea, solamente questo corso dedicato all'organizzazione delle macchine numeriche. Tuttavia fornisce anche le basi necessarie per seguire eventuali corsi più avanzati e dedicati al progetto di sistemi di elaborazione da un punto di vista dell'hardware.</p> <p>Al termine del corso, gli studenti conosceranno le basi della programmazione assembly (RISC-V, ARM e Intel) e i concetti principali delle architetture dei calcolatori, con particolare riferimento ai sistemi RISC-V.</p>
Calculus 1	<p>1. In-depth knowledge of the basic methods of calculus for functions of one variable, as specified in the course program. Understanding the main characteristics of differentiable and integrable real-valued functions. Ability to apply the methods to concrete problems and other scientific or industrial problems.</p> <p>2. Ability to use deductive reasoning in dealing with problems concerning the analysis of functions. Ability to set up a problem and solve it using appropriate analysis tools.</p> <p>3. Autonomy of judgment Ability to develop logical arguments, to understand when a proof is rigorous and complete, and to extend and produce minor extensions of the proofs seen in the course. Ability to identify the most appropriate methods for analyzing and interpreting problems.</p> <p>4. Learning skills Ability to acquire and manage new information starting from the statements and proofs developed in the lectures.</p>
Calculus 2	<p>1. Knowledge and ability to understand.</p> <p>Thorough knowledge of the basic topics of differential and integral calculus for functions of several variables as specified in the course syllabus. Students are expected to solve exercises on topics covered in the course, express concepts introduced during lectures in appropriate mathematical language, and provide correct mathematical proofs.</p> <p>2. Ability to apply knowledge and understanding.</p> <p>Students should acquire inductive and deductive reasoning skills in dealing with</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>problems and the ability to schematize a physical or geometric phenomenon in rigorous terms.</p> <p>3. Autonomy of judgment</p> <p>Students should be able to develop simple logical arguments, recognize the correctness of simple proofs, and produce correct proofs. Moreover, students must be able to identify the most appropriate methods for analyzing and addressing problems solvable with the techniques acquired.</p> <p>4. Communication skills</p> <p>Students should be able to present the topics related to mathematical analysis with precision, accurately stating the definitions and theorems given in the lectures' register, and presenting flawless proofs supported by solid logical reasoning.</p> <p>5. Learning skills</p> <p>Students should acquire and handle new information concerning inherent in techniques and problems related to mathematical analysis. Finally, students are expected to acquire a more solid and global understanding of the fundamental concepts studied in the courses Calculus 1, and Geometry and Linear Algebra.</p>
Computational Logic	<p>The student can describe the basic concepts and tools in computational logic and use them to reason about computer science and AI applications.</p> <p>The student can build logical models for various computational problems.</p>
Computer Architectures	<p>The goal of this course is to provide the basic notions about the organization and architecture of computing machines, either general-purpose computers, or specialized architectures designed to handle specific tasks. Furthermore, the course provides basic knowledge about the assembly language (in particular using as a reference RISC-V and briefly discussing the Intel and ARM architectures), the HW/SW interface, and the low-level mechanisms of a computer. Finally, the course also provides the fundamental knowledge needed to attend advanced courses focused on the design of computer systems from the hardware perspective.</p> <p>This course does not require any preliminary knowledge. The structure of the course is especially suited for students with a CV oriented towards software and systems,</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>who during their Bachelor's Degree have the chance to learn about computer architectures only through this course. On the other hand, the course will also provide the fundamental knowledge needed to attend advanced courses focused on the design of computer systems from the hardware perspective.</p> <p>At the end of the course, students will know the basics of assembly programming (RISC-V, ARM, and Intel) and the main concepts of computer architectures, with particular reference to RISC-V systems.</p>
Computer Programming 1	<p>Learning Objectives</p> <p>Introduce the student to algorithms, code programming, programming principles and techniques.</p> <p>Make the student familiar with code programming and provide the basic notions, insights and tools of code programming</p> <p>Learn fundamentals of imperative programming</p> <p>Learn program structures, elements, and operations</p> <p>Learn by doing: learn how to develop programs to face problems of simple/average difficulty</p> <p>Learn the (imperative) language C++</p> <p>Learn introductive notions of object-oriented programming in C++</p> <p>Expected Learning Outcome</p> <p>At the end of the course the student (1) will know and understand the fundamental notions and concepts about the imperative programming and about the C++ language and (2) will be able to analyze problems of simple/average difficulty and (3) will be able to elaborate code solutions (i.e., develop software programs), by using C++ as imperative programming language to face them.</p>
Computer Programming 2	<p>The course aims to provide fundamental concepts characterizing object-oriented programming (OOP): class, object, inheritance, polymorphism, information hiding, static and dynamic binding. Course attendance will lead to the required competences to correctly structure a program using OOP, identifying and implementing its constituent classes, using a basic UML representation that will be deepened in</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	subsequent courses and documenting the code. The language used will be mainly Java, but frequent references will be made to other OOP languages, especially C ++.
Databases	In today's data-driven world, almost every human activity generates or relies on data. Efficient solutions for storing, managing, searching, and retrieving large volumes of data are essential. This course aims to provide students with an understanding of data management systems, with a strong focus on relational databases, which continue to dominate the industry.
Fisica	<p>Obiettivi di base</p> <p>Comprensione dei fondamenti della fisica: Fornire una solida base nei principi fondamentali della fisica (meccanica, termodinamica, elettromagnetismo, ottica e fisica moderna) essenziali per comprendere fenomeni naturali e processi tecnologici.</p> <p>Matematizzazione dei fenomeni fisici: Sviluppare la capacità di descrivere i fenomeni fisici attraverso modelli matematici e strumenti analitici, indispensabili per la programmazione di simulazioni fisiche.</p> <p>Obiettivi trasversali</p> <p>Pensiero critico e problem-solving: Promuovere il pensiero critico e la capacità di risolvere problemi complessi attraverso un approccio scientifico, utile sia per la fisica che per l'informatica.</p> <p>Integrazione interdisciplinare: Preparare gli studenti a lavorare in team interdisciplinari, dove l'informatica si applica a problemi legati alla fisica, come nel machine learning per l'analisi di dati fisici o nello sviluppo di algoritmi per la meccanica quantistica computazionale.</p>
Fondamenti di elaborazione dei segnali	Il corso fornirà conoscenze su segnali deterministici mono e multi-dimensional, in formato analogico e numerico. Si apprenderà ad analizzare un segnale nel suo dominio originale e nel dominio della frequenza, tramite opportune trasformazioni. Si vedrà anche come un segnale viene modificato da un sistema di elaborazione, nel continuo e nel discreto, e come progettare alcuni semplici sistemi per realizzare alcune funzioni basilari.

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

Fondamenti matematici per l'informatica	<p>L'insegnamento si propone di fornire allo studente alcune nozioni di base relative all'aritmetica modulare e alla teoria dei grafi.</p> <p>RISULTATI DI APPRENDIMENTO ATTESI</p> <p>Al termine dell'insegnamento, lo studente sarà in grado:</p> <p>(a)_1 di presentare e dimostrare alcuni risultati fondamentali dell'aritmetica modulare e della teoria dei grafi;</p> <p>(a)_2 di applicare i suddetti risultati per risolvere problemi riguardanti il principio di induzione e le congruenze;</p> <p>(a)_3 di applicare i suddetti risultati per risolvere problemi riguardanti i grafi.</p>
Formal Language and compilers	<p>The course introduces the techniques and the algorithms for the design and for the implementation of interpreters and of compilers of programming languages.</p> <p>Nonetheless, the methodologies learnt in the course are at the basis of various applications in areas other than the compilation of formal languages (e.g., analysis of natural languages; analysis of the soundness of queries).</p>
Functional Programming	<p>The course describes and analyses the characteristics of various programming languages, introduces paradigms of programming, and studies in detail the functional paradigms (with reference to the language ML).</p>
Fundamentals of signal processing	<p>The didactic target of this module is to introduce the student to the field of signal engineering. The course will start from the theoretical and mathematical concept of signal, expressed in the different domains. Then the analysis and representation of signals in the frequency domain will be propaedeutic to the actual processing of the signals. Signal processing will be introduced by the fundamental concepts of distortion and filtering in the analogic framework. On the other hand, the final part of the course will reconsider the fundamental concepts of signal processing in the mono and multidimensional numerical framework. At the end of the course, the student will be able to analyze signals and their processing both in the time domain and in the frequency domain and to design simple analog and numerical filtering systems.</p>
Geometria e Algebra Lineare	<p>OBIETTIVI FORMATIVI</p> <p>L'insegnamento si propone di fornire allo studente alcune nozioni di base relative</p>



REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>all'aritmetica modulare e alla teoria dei grafi.</p> <p>RISULTATI DI APPRENDIMENTO ATTESI</p> <p>Al termine dell'insegnamento, lo studente sarà in grado:</p> <p>(a)_1 di presentare e dimostrare alcuni risultati fondamentali dell'aritmetica modulare e della teoria dei grafi;</p> <p>(a)_2 di applicare i suddetti risultati per risolvere problemi riguardanti il principio di induzione e le congruenze;</p> <p>(a)_3 di applicare i suddetti risultati per risolvere problemi riguardanti i grafi.</p>
Geometry and linear Algebra	<p>The course aims to provide the basic elements of Linear Algebra, with applications to the geometry in three-dimensional space and to vector calculus.</p> <p>At the end of the course the student should:</p> <p>have a good knowledge and understanding of the theory of finite dimensional real vector spaces, as described in the course list of topics;</p> <p>know the basic concepts, be able to explain and describe them also with examples, identify the connections between them;</p> <p>be familiar with the mathematical tools which are needed to develop a suitable approach and solve Linear Algebra problems, such as, for example, operations between matrices, echelon form reduction algorithms, computation of determinants, of eigenvalues and eigenvectors.</p>
Human Computer Interaction	<p>The course has the general objective of providing students with the fundamental notions about the interaction between humans and computerized systems. In particular, the course is focused on the principles of the User-Centered-Design methodology, as well as on their application during the design and evaluation of interfaces of various types and for various domains.</p>
Ingegneria del Software	<p>Un approccio ingegneristico allo sviluppo del software è fondamentale per far fronte alla crescente complessità dei sistemi e per garantire la loro sostenibilità economica. Come per ogni altra disciplina ingegneristica, la produzione di software richiede sistematicità delle attività, l'adozione di processi ben definiti e strutturati e l'utilizzo di metodi e tool di supporto in grado di garantire qualità e il rispetto dei vincoli di</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>progetto. L'obiettivo del corso è quello di fornire agli studenti concetti, approcci e tecniche per l'analisi, la progettazione e lo sviluppo del software. In particolare, il corso affronterà tematiche legate ai modelli di processo di sviluppo, linguaggi di modellazione e tecniche di verifica e validazione. Verranno introdotti linguaggi diagrammatici di modellazione, metodi di analisi dei requisiti, principi di progettazione e tecniche di testing e validazione che permetteranno allo studente di cimentarsi nell'analisi e nella progettazione di software di qualità.</p> <p>Sono previsti laboratori pratici in cui si andrà a progettare e implementare un servizio web applicando i più moderni metodi di sviluppo, incluso l'adozione delle direttive RESTful e l'uso di linguaggi standard per la documentazione delle web APIs, la collaborazione nello sviluppo del codice sorgente tramite un repository centralizzato, l'automazione del testing del codice tramite la definizione di suite di test, e la configurazione di vari servizi e tool per instaurare un processo di integrazione continua del software. Verranno infine introdotti concetti project management e metriche di prodotto per il software.</p> <p>RISULTATI DI APPRENDIMENTO ATTESI.</p> <p>Al termine del percorso, lo studente sarà in grado di:</p> <ol style="list-style-type: none"> 1. definire il processo e il modello di sviluppo più appropriato per l'analisi e la progettazione del software in esame; 2. rappresentare attraverso le diverse tipologie di diagrammi i requisiti, l'architettura, le interazioni fra i diversi componenti e il deployment del sistema; 3. definire e gestire in maniera consistente e completa i requisiti funzionali e non funzionali di un sistema software individuandone priorità e criticità; 4. definire l'architettura di un sistema software adottando stili architetturali e principi di raffinamento, astrazione, modularità, indipendenza funzionale e refactoring; 5. progettare un componente di un sistema software identificandone il tipo di controllo e seguendo il processo step-by-step per il rispetto dei principi generali di progettazione; 6. definire e documentare un servizio web nel rispetto delle direttive RESTful,
--	---

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	collaborare allo sviluppo del codice su un repository centralizzato, testarlo, e “deployare” il software tramite un processo continuo e automatizzato.
Introduction to Computer and Network Security	Introductory course to the discipline of computer and network security. The main objectives are to enable students to (i) understand the theoretical and practical problems of information security, (ii) understand and recognize threats to the fundamental properties of security: confidentiality, integrity, and availability - and (iii) understand how key security mechanisms (such as cryptography, authentication and authorization protocols, and access control) can be applied to mitigate vulnerabilities.
Introduction to Machine Learning	This course provides a broad introduction to machine learning and statistical pattern recognition. The course will cover different topics ranging from supervised learning (parametric/non-parametric algorithms, kernel methods, feedforward neural networks), unsupervised learning (clustering, dimensionality reduction, autoencoders, deep generative models) and reinforcement learning. In the final part of the course some applications of machine learning will be presented in the field of computer vision, robotics and natural language processing. Theoretical discussion will be complemented with lab in Python using open-source libraries.
Introduzione alla Programmazione per il Web	<p>L'insegnamento si propone di fornire allo studente / studentessa la comprensione della web e degli strumenti necessari per costruire delle web applications.</p> <p>Alla fine del corso gli studenti / studentesse:</p> <p>conosceranno i concetti fondamentali che caratterizzano la web: protocollo http, linguaggi html, javascript e css, ciclo request-response, framework basati su asynchronous javascript, persistenza dello stato, interazione con database;</p> <p>utilizzeranno i più comuni linguaggi tipici della web ed avranno familiarizzato con il concetto di DOM;</p> <p>saranno in grado di progettare e sviluppare semplici web applications.</p> <p>La conoscenza acquisita permetterà agli studenti / studentesse di autoapprendere in futuro con poco sforzo qualsiasi framework per la progettazione e gestione di sistemi web.</p>
Laboratorio di	Il corso si prefigge di presentare in generale gli elementi fondamentali della

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

programmazione per sistemi mobili e tablet	<p>programmazione di sistemi mobili, e di fornire conoscenze e competenze necessarie per la progettazione e lo sviluppo di applicazioni.</p> <p>Si articola nelle seguenti sezioni:</p> <ol style="list-style-type: none"> 1) Introduzione alla programmazione di dispositivi mobili 2) Analisi critica dei requisiti di una mobile application in relazione agli utenti target 3) Le componenti fondamentali: activities, tasks, processes, services, intents, broadcast receivers, content providers, fragments. Widgets, views e layouts, async tasks, adapters, interazione con databases e files, uso di sensori 4) Rapid prototyping: wireframing 5) Principi di user interface, user interaction e product design
Linguaggi Formali e Compilatori	<p>Lo scopo del corso e' introdurre le nozioni fondamentali dei linguaggi formali e le basi della progettazione di compilatori di linguaggi di programmazione.</p> <p>I principi e le tecnologie della compilazione trovano numerose applicazioni in aree ortogonali a quella dei linguaggi di programmazione (e.g., analisi di linguaggi naturali; analisi e controllo di correttezza sintattica di query).</p> <p>Lo studente apprendera' tecniche ed algoritmi per la progettazione del front-end di compilatori e per la gestione e progettazione di applicazioni per l'analisi e la trasformazione di testi.</p>
Logic networks	<p>The course aims at providing students with the fundamental notions of digital circuits, with particular reference to the understanding, analysis and design of logic networks.</p> <p>At the end of the course, the student will be able to understand the functioning of digital circuits making use of logic gates and state machines, and to design complex digital processing systems by analyzing and evaluating the effect of design choices.</p>
Logica Computazionale	<p>Il corso mira a fornire le conoscenze di base della logica, intesa come un linguaggio formale per il modellamento della conoscenza e del ragionamento. Il corso copre le logiche più importanti con applicazioni nei sistemi di ragionamento automatico, Intelligenza Artificiale, Metodi Formali, e la modellazione e l'utilizzo dei dati del Web.</p>
Mathematics for Informatics	<p>OBJECTIVES</p> <p>To provide an understanding of modular arithmetic and graph theory.</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>LEARNING OUTCOMES</p> <p>Students will be able:</p> <p>(a)_1 To present and to prove some basic results concerning modular arithmetic and graph theory;</p> <p>(a)_2 To apply the above-mentioned results to solve problems concerning induction and congruences;</p> <p>(a)_3 To apply the above-mentioned results to solve problems concerning graphs.</p>
Networking	<p>This course covers the basic topics of telecommunications networks, providing the student with the knowledge necessary to understand how communication between devices works in local area networks and the Internet.</p> <p>Upon completion of the course, students will be able to:</p> <ul style="list-style-type: none"> - Understand the basic concepts of telecommunication networks - Discuss the operating principles of various network protocols - Predict the behavior of a congestion control algorithm - Solve routing exercises - Design simple IP networks
Operating Systems	<p>The aim of the course is to provide a solid understanding of the fundamental concepts that underlie modern operating systems, such as concurrency, memory management, and file systems. Each topic will be addressed from a theoretical perspective, and select topics will be illustrated with practical examples.</p> <p>By the end of the course, students will have gained the ability to explain the architecture of an operating system, understand how and why its core features are designed and implemented, and critically evaluate different design solutions. Through practical lessons, students will gain hands-on experience with operating system shells (covering both basic and advanced commands and scripting), as well as with advanced topics like threads, synchronization, and concurrency.</p>
Optimization techniques	<p>The goal of this course is to provide students a solid basis in mathematical optimization and on classical optimization algorithms.</p>
Probabilità e statistica	<p>L'obiettivo formativo del corso è di introdurre gli studenti al Calcolo delle Probabilità e</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>alla Statistica. Queste sono due discipline fondamentali in una laurea in Informatica, poiché costituiscono la base per tecniche utilizzate in molti campi dell'Intelligenza Artificiale, del Machine Learning, dello Statistical Learning e della Data Science.</p>
Probability	<p>1. Knowledge and Understanding In-depth knowledge of the basic topics in probability and statistics as specified by the course syllabus. Ability to identify a probabilistic model and understand its main characteristics. Ability to apply probability to physics and other scientific or common life problems.</p> <p>2. Applying Knowledge and Understanding Ability to use inductive and deductive reasoning to address problems involving random phenomena. Capability to formalize a random phenomenon in rigorous terms, set up a problem, and solve it using appropriate tools from probability and statistics.</p> <p>3. Autonomy of Judgment Ability to develop logical arguments and provide correct proofs. Skill to identify the most suitable methods for analyzing and interpreting problems.</p> <p>4. Communication Skills Ability to present topics related to probabilistic and statistical nature.</p> <p>5. Learning Ability Capability to acquire and manage new information related to models involving randomness.</p>
Programmazione 1	<p>L'obiettivo del corso è l'apprendimento dei fondamenti della programmazione imperativa. Come linguaggio verrà usato (un sottoinsieme stretto del) C++, visto prettamente come linguaggio di programmazione imperativo.</p>
Programmazione 2	<p>L'insegnamento si propone di fornire allo studente la comprensione dei paradigmi di programmazione object-oriented e event-based. Verranno acquisiti i concetti fondamentali che caratterizzano ogni linguaggio di programmazione (gestione della memoria, regole di scope, visibilità degli identificatori, codice strutturato e strutture dati, astrazione), e quelli che caratterizzano la programmazione orientata agli oggetti (OOP): classe, oggetto, ereditarietà, polimorfismo, information hiding, binding statico e</p>

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>dinamico. Lo studente verrà anche introdotto alla programmazione funzionale (FP). Al termine del corso, lo studente sarà in grado di identificare e descrivere le caratteristiche dei diversi paradigmi e scegliere quello più adatto alla soluzione di specifici problemi. Sarà inoltre in grado di strutturare correttamente un programma usando OOP, identificandone e implementandone le classi costituenti, utilizzando una rappresentazione UML di base che verrà approfondita in corsi successivi e documentando il codice. Sarà capace di utilizzare i linguaggi Standard ML e Java, ed avrà familiarizzato con le API di quest'ultimo. Avrà anche una conoscenza di base dei costrutti OOP del linguaggio C++. La conoscenza acquisita permetterà allo studente di autoapprendere in futuro con poco sforzo altri linguaggi (es. Python e C#).</p>
Programmazione avanzata	<p>Lo studente comprenderà e applicherà astrazioni e caratteristiche di un moderno linguaggio multiparadigma orientato all'efficienza, ovvero gli standard recenti del C++.</p>
Programmazione Funzionale	<p>Obiettivo formativo del corso è quello di fornire agli studenti/studentesse i concetti fondamentali dei linguaggi e dei paradigmi di programmazione e, in particolare, di introdurre gli studenti/studentesse al paradigma di programmazione funzionale. In dettaglio, il corso introdurrà alcune caratteristiche dei linguaggi e paradigmi di programmazione utili a comprendere il paradigma funzionale. Il corso si focalizzerà in particolare sul linguaggio ML e offrirà una introduzione del linguaggio Scala. Il corso prevede laboratori pratici in cui gli studenti/studentesse potranno esercitarsi con esercizi e task di programmazione funzionale in linguaggio ML.</p> <p>La frequenza e la partecipazione attiva alle due attività formative proposte dal corso (lezioni frontali, laboratori pratici) e lo studio e l'esercizio individuale consentiranno a studenti/studentesse di:</p> <ul style="list-style-type: none"> conoscere i concetti fondamentali che caratterizzano i linguaggi di programmazione (regole di scope, visibilità degli identificatori, strutture di controllo e strutture dati) comprendere le caratteristiche principali dei diversi paradigmi di programmazione ed in particolare del paradigma di programmazione funzionale applicare le conoscenze acquisite su task ed esercizi sviluppare soluzioni nel linguaggio di programmazione funzionale ML.

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

Reti	Diamo tutti per scontato che il nostro smartphone possa accedere ad una story sui social, che i nostri messaggi e le nostre foto arrivino a destinazione, che le videochiamate con famiglia e amici funzionino alla pressione di un tasto. Ma nessuna di queste cose potrebbe esistere senza la (complessa) infrastruttura di telecomunicazione che permette loro di scambiarsi dati: Internet.
Reti logiche	L'insegnamento si propone di fornire agli studenti le nozioni fondamentali dei circuiti digitali, con particolare riferimento alla comprensione, l'analisi e la progettazione di reti logiche. Al termine dell'insegnamento, lo studente sarà in grado di comprendere il funzionamento di circuiti digitali facenti uso di porte logiche e di macchine a stati, e di progettare sistemi di elaborazione digitali complessi analizzando e valutando l'effetto delle scelte progettuali. Lo studente impara ad affrontare il progetto attraverso la specifica tramite l'uso di modelli sequenziali sincroni, l'analisi tramite diagrammi, e la simulazione e l'implementazione con il linguaggio VHDL, identificando le componenti critiche, e partizionando il progetto per gestirne al meglio la complessità. Esercitazioni su problemi di progetto ed esperienze di laboratorio consentono allo studente di applicare nella pratica i concetti teorici, e di acquisire familiarità con componenti e sistemi di sviluppo reali.
Sistemi Informativi	Il corso si propone di fornire le competenze necessarie a comprendere come le tecnologie dell'informazione e della comunicazione (ICT) possono essere usate in diversi tipi di organizzazione per gestire i propri processi di business. Nel corso saranno affrontati temi quali: <ul style="list-style-type: none"> - Cos'è un SIA? Come si colloca all'interno di un'azienda? - Cos'è l'«esigenza informativa» di un'azienda? - Come si possono classificare i diversi tipi di SIA? - Quali sono i principali «moduli» operazionali di un SIA? - Cosa significa gestire un SIA dal punto di vista del data center aziendale? - In che modo nuove tecnologie (come Big Data, IA, Internet of Things, ...) possono impattare sulle organizzazioni del futuro?
Sistemi Operativi	Obiettivo dell'insegnamento è quello di fornire i concetti fondamentali che sono alla

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>base dei moderni sistemi operativi, con particolare enfasi sulla gestione della concorrenza, della memoria e del file system. Ogni concetto sarà trattato dal punto di vista teorico. Per i concetti più importanti la trattazione riguarderà anche aspetti più pratici in modo da familiarizzare con loro la complessità realizzativa.</p> <p>Al termine dell'insegnamento, lo studente sarà in grado di comprendere l'architettura di un sistema operativo, di capire come sono progettate e implementate le principali funzionalità di un sistema operativo, di confrontare e valutare le diverse soluzioni progettuali e implementative.</p>
Software Engineering	<p>General Aspects</p> <p>Regarding the aspect of "Knowledge and understanding" the student will be able to: know the fundamentals, techniques and methods of design, customization and implementation of software to support the automation of new-generation information systems for industrial production and business;</p> <p>Regarding the aspect of "Applying knowledge and understanding" the student will be able to:</p> <p>design and perform experimental analyses of information systems in order to acquire measures related to their behaviour and to evaluate experimental hypotheses in different fields of application, such as business, industrial or research;</p> <p>define an innovative technical solution to an application problem that meets technical, functional and organizational constraints and requirements;</p> <p>Regarding the aspect of "Making judgments" the student will be able:</p> <p>to plan and re-plan a technical project activity and carry it out in accordance with defined deadlines and objectives;</p> <p>to reconcile the objectives of the project that are in conflict, to trade off cost, resources, time, knowledge or risk.</p> <p>Regarding the aspect of "Communication skills" the student will be able:</p> <p>to coordinate project teams and to identify activities to achieve project objectives;</p> <p>to present the contents of a scientific/technical report to an audience, including non-specialists, at a fixed time;</p>



REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

	<p>to prepare and conduct technical presentations in English;</p> <p>to carry out research and projects in a collaborative manner;</p> <p>Regarding the aspect of "Learning skills" the student will be able:</p> <p>to keep up to date independently with developments in the most important areas of information technology;</p> <p>extend the knowledge acquired during the course of study by reading and Understanding scientific and technical documentation in English.</p> <p>Specific objectives</p> <p>At the end of the class, the student will be able to:</p> <p>design the most suitable development process and model for the analysis and the design of the software under consideration;</p> <p>represent by using different types of diagrams the requirements, the architecture, the interaction among components, entities and the deployment of the system;</p> <p>define and manage the functional as well as non functional requirements of a software system by highlighting priorities and potential issues;</p> <p>design the architecture of a software system by adopting architectural styles and principles of refinement, abstraction, modularity, functional independence, and refactoring;</p> <p>describe requirements, and the components and implement them in an agile manner, with short development cycles to ensure periodic and frequent delivery of software to users;</p> <p>design, implement and describe a RESTful compliant web service, collaborate on the code development test it, and "deploy" the software through a continuous, automated process.</p>
--	---

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

Allegato 2: ARTICOLAZIONE DEL CORSO

Le tabelle sotto riportate costituiscono le tabelle allegate al Regolamento didattico del corso di Laurea in Informatica, attivato nella Classe L-31, Classe delle lauree in SCIENZE E TECNOLOGIE INFORMATICHE, di cui al DM 19/12/2023 n 1648 ed è conforme a quanto previsto dall'Ordinamento didattico e si applicano a partire dalla coorte 2025/2026. Il corso di Laurea Triennale in 'Informatica' si articola in due curricula:

- 1) Scienze e Tecnologie Informatiche (in lingua italiana),
- 2) Computer Science (in lingua inglese),

della durata normale di 3 anni, per un totale di 180 CFU.

Tutti i percorsi prevedono in comune:

- 24 crediti nei settori MAT/* per la **formazione matematico-fisica di base**
- 18 crediti nei settori INF/01, ING-INF/05 per la **formazione informatica di base**
- 12 crediti nei **settori affini** indicati nell'ordinamento
- 60 crediti nei **settori caratterizzanti** INF/01, ING-INF/05

Tutti i percorsi continuano con:

- Ulteriori 6 crediti nei **settori affini** indicati nell'ordinamento
- Ulteriori 12 crediti nei **settori caratterizzanti** INF/01, ING-INF/05 o affini indicati nell'ordinamento
- Ulteriori 18 crediti nei **settori caratterizzanti** INF/01, ING-INF/05

Tutti i percorsi si completano con:

- Attività formative a **scelta dello/a studente/essa** pari a 12 crediti
- La prova per la conoscenza della **lingua** (Inglese Livello B1 per i percorsi in italiano, Inglese livello C1, o Inglese Tecnico, o conoscenza base di Italiano per studenti e studentesse stranieri, nel caso del percorso in inglese) pari a 3 crediti;
- Il **tirocinio o internato** formativo pari a 9 crediti è un'esperienza professionalizzante che permette allo/a studente/essa di approfondire le conoscenze apprese nel corso degli studi universitari, di orientare le sue future scelte professionali e di studiare il possibile trasferimento tecnologico delle attività di ricerca e sviluppo tecnologico effettuate durante la tesi di laurea. Esso consiste in un periodo di formazione svolto presso enti, aziende, studi professionali o istituzioni a complemento od integrazione del percorso di studio.
- La **prova finale**, pari a **6 crediti**, riporta l'attività di formazione svolta dallo studente sotto la guida di un docente o ricercatore dell'università o titolare di un insegnamento offerto in un Corso di Studio organizzato dal Dipartimento, come specificato nel "Regolamento prova finale e conferimento del titolo", pubblicato nel sito del Dipartimento.

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA
Tabella 2. Articolazione del Percorso SCIENZE E TECNOLOGIE INFORMATICHE (in lingua italiana)

NOME INSEGNAMENTO	CFU	SSD	SSD DM 639/2024	TIPO ATTIVITÀ FORMATIVA	ANNO	PROPEDEU TICITÀ
Insegnamenti obbligatori: 114 CFU						
Analisi matematica 1	12	MAT/05	MATH-03/A	Base	1	---
Geometria e Algebra Lineare	6	MAT/03	MATH-02/B	Base	1	---
Programmazione 1	12	ING-INF/05	IINF-05/A	Base	1	---
Calcolatori	6	ING-INF/05	IINF-05/A	Base	1	---
Fondamenti matematici per l'informatica	6	MAT/03	MATH-02/B	Base	1	---
Probabilità e statistica	6	MAT/06	MATH-03/B	Affine	1	---
Programmazione 2	6	INF/01	INFO-01/A	Caratteriz zante	1	---
Programmazione Funzionale	6	INF/01	INFO-01/A	Caratteriz zante	1	---
Algoritmi e strutture dati - parte prima	6	INF/01	INFO-01/A	Caratteriz zante	2	---
Algoritmi e strutture dati - parte seconda	6	INF/01	INFO-01/A	Caratteriz zante	2	---
Basi di dati	6	ING-INF/05	IINF-05/A	Caratteriz zante	2	---
Reti	6	INF/01	INFO-01/A	Caratteriz zante	2	---
Ingegneria del Software	12	ING-INF/05	IINF-05/A	Caratteriz zante	2	---



REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

NOME INSEGNAMENTO	CFU	SSD	SSD DM 639/2024	TIPO ATTIVITÀ FORMATIVA	ANNO	PROPEDEU TICITÀ
Sistemi Operativi	12	ING-INF/05	IINF-05/A	Caratteriz zante	2	---
Logica Computazionale	6	MAT/01	MATH-01/A	Affine	3	---

NOME INSEGNAMENTO	CFU	SSD	SSD DM 639/2024	TIPO ATTIVITÀ FORMATIV A	ANNO	PROPEDEU TICITÀ
Insegnamenti obbligatori: 24 CFU						
Fisica	6	FIS/01	PHYS-01/A	Affine	2	---
Linguaggi Formali e Compilatori	12	INF/01	INFO-01/A	Caratteri zzante	3	---
Introduction to Machine Learning	6	INF/01	INFO-01/A	Caratteri zzante	3	---

NOME INSEGNAMENTO	CFU	SSD	SSD DM 639/2024	TIPO ATTIVITÀ FORMATIVA	ANNO	PROPEDEU TICITÀ
Insegnamenti a scelta vincolata: 12 CFU						
Introduction to Computer and Network Security	6	INF/01	INFO-01/A	Caratteriz zante	2	---
Sistemi Informativi	6	INF/01	INFO-01/A	Caratteriz zante	2	---

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

NOME INSEGNAMENTO	CFU	SSD	SSD DM 639/2024	TIPO ATTIVITÀ FORMATIVA	ANNO	PROPEDEU TICITÀ
Reti logiche	6	ING-INF/01	IINF-01/A	Affine	2	---
Introduzione alla Programmazione per il Web	6	ING-INF/05	IINF-05/A	Caratteriz zante	2	---
Analisi matematica 2	6	MAT/05	MATH-03/A	Affine	2	---
Fondamenti di elaborazione dei segnali	6	ING-INF/03	IINF-03/A	Affine	3	---
Programmazione avanzata	6	ING-INF/05	IINF-05/A	Caratteriz zante	3	---
Optimization techniques	6	MAT/09	MATH-06/A	Affine	3	---
Laboratorio di programmazione per sistemi mobili e tablet	6	INF/01	INFO-01/A	Caratteriz zante	3	---
Human Computer Interaction	6	ING-INF/05	IINF-05/A	Caratteriz zante	3	---

Attività obbligatorie: 30 CFU	
Corsi a scelta libera	12
Tirocinio	9
Inglese (livello B1)	3
Prova finale	6
Totale CFU	180

REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA
TABELLA 2: Articolazione DEL PERCORSO COMPUTER SCIENCE (in lingua inglese)

NOME INSEGNAMENTO	CFU	SSD	SSD DM 639/2024	TIPO ATTIVITÀ FORMATIVA	ANNO	PROPEDEUT ICITÀ
Insegnamenti obbligatori: 114 CFU						
Calculus 1	12	MAT/05	MATH-03/A	Base	1	---
Geometry and Linear Algebra	6	MAT/03	MATH-02/B	Base	1	---
Computer Programming 1	12	ING-INF/05	IINF-05/A	Base	1	---
Computer Architectures	6	ING-INF/05	IINF-05/A	Base	1	---
Mathematics for Informatics	6	MAT/05	MATH-03/A	Base	1	---
Probability	6	MAT/06	MATH-03/B	Affine	1	---
Computer Programming 2	6	INF/01	INFO-01/A	Caratteriz zante	1	---
Functional Programming	6	INF/01	INFO-01/A	Caratteriz zante	1	---
Algorithms and Data Structures – part 1	6	INF/01	INFO-01/A	Caratteriz zante	2	---
Algorithms and Data Structures – part 2	6	INF/01	INFO-01/A	Caratteriz zante	2	---
Databases	6	ING-INF/05	IINF-05/A	Caratteriz zante	2	---
Networking	6	INF/01	INFO-01/A	Caratteriz zante	2	---
Computational Logic	6	MAT/01	MATH-01/A	Caratteriz zante	3	---
Software Engineering	12	ING-INF/05	IINF-05/A	Affine	2	---
Operating Systems	12	ING-INF/05	IINF-05/A	Caratteriz zante	2	---



REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

NOME INSEGNAMENTO	CFU	SSD	SSD DM 639/2024	TIPO ATTIVITÀ FORMATIVA	ANNO	PROPEDEU TICITÀ
Insegnamenti obbligatori: 24 CFU						
Optimization Techniques	6	MAT/09	MATH-06/A	Affine	3	---
Formal Language and compilers	12	INF/01	INFO-01/A	Caratteriz zante	3	---
Introduction to Machine Learning	6	INF/01	INFO-01/A	Caratteriz zante	3	---

NOME INSEGNAMENTO	CFU	SSD	SSD DM 639/2024	TIPO ATTIVITÀ FORMATIVA	ANNO	PROPEDEU TICITÀ
Insegnamenti a scelta vincolata : 12 CFU						
Introduction to Computer and Network Security	6	INF/01	INFO-01/A	Caratteriz zante	2	---
Fundamentals of signal processing	6	ING-INF/03	IINF-03/A	Affine	2	---
Logic networks	6	ING-INF/01	IINF-01/A	Affine	2	---
Calculus 2	6	MAT/05	MATH-03/A	Affine	2	---
Advanced Programming	6	ING-INF/05	IINF-05/A	Caratteriz zante	3	---
Human Computer Interaction	6	ING-INF/05	IINF-05/A	Caratteriz zante	3	---

Attività obbligatorie: 30 CFU		
Per gli studenti di lingua italiana, a scelta tra:		
Technical English	3	
Inglese (livello C1)		
Per gli studenti di lingua straniera:		



REGOLAMENTO DIDATTICO DEL CORSO DI LAUREA IN INFORMATICA

Prova di conoscenza lingua italiana	3
Corsi a scelta libera	12
Tirocinio	9
Prova finale	6
Totale CFU	180