

# Syllabus

## *Descrizione corso*

<b>Titolo insegnamento</b>	Data Structures & Algorithms
<b>Codice insegnamento</b>	76410
<b>Titolo aggiuntivo</b>	
<b>Settore Scientifico-Disciplinare</b>	INFO-01/A
<b>Lingua</b>	Italiano
<b>Corso di Studio</b>	Corso di laurea in Informatica e Management delle Aziende digitali
<b>Altri Corsi di Studio (mutuati)</b>	
<b>Docenti</b>	dr. Ivan Donadello, Ivan.Donadello@unibz.it <a href="https://www.unibz.it/en/faculties/engineering/academic-staff/person/45237">https://www.unibz.it/en/faculties/engineering/academic-staff/person/45237</a>
<b>Assistant</b>	
<b>Semestre</b>	Primo semestre
<b>Anno/i di corso</b>	2
<b>CFU</b>	6
<b>Ore didattica frontale</b>	40
<b>Ore di laboratorio</b>	20
<b>Ore di studio individuale</b>	90
<b>Ore di ricevimento previste</b>	
<b>Sintesi contenuti</b>	<ul style="list-style-type: none"><li>- Ricerca e ordinamento</li><li>- Analisi degli algoritmi: correttezza e complessità</li><li>- Dividere e conquistare, ricorsività</li><li>- Puntatori, strutture di dati dinamiche, liste collegate</li><li>- Tipi di dati astratti: pile, code, code di priorità, mappe</li><li>- Algoritmi elementari a grafo e ad albero</li></ul>
<b>Argomenti dell'insegnamento</b>	<ul style="list-style-type: none"><li>- Ricerca e ordinamento</li><li>- Analisi degli algoritmi: correttezza e complessità</li><li>- Divide and conquer, ricorsività</li></ul>

	<ul style="list-style-type: none"> <li>- Puntatori, strutture dati dinamiche, liste collegate</li> <li>- Tipi di dati astratti: pile, code, code di priorità, mappe</li> <li>- Algoritmi elementari su grafi e alberi</li> </ul>
<b>Parole chiave</b>	Ricerca e ordinamento; Strutture dati e algoritmi; Analisi degli algoritmi; Ricorsività, Graph algorithms
<b>Prerequisiti</b>	<ul style="list-style-type: none"> <li>- Competenze di programmazione Java/C a livello introduttivo</li> <li>- Conoscenze matematiche di base su insiemi, relazioni, matematica discreta, funzioni matematiche (logaritmi ed esponenziali), serie e calcolo differenziale.</li> </ul>
<b>Insegnamenti propedeutici</b>	
<b>Modalità di insegnamento</b>	Lezioni frontali e laboratori.
<b>Obbligo di frequenza</b>	<p>Non è obbligatoria, ma fortemente consigliata.</p> <p>Le lezioni consistono in presentazioni, intervallate da piccoli esercizi e discussioni con gli studenti.</p> <p>L'obiettivo è mettere gli studenti in grado di sviluppare e analizzare algoritmi, un'abilità che può essere acquisita solo con l'allenamento.</p> <p>Gli studenti che non possono seguire tutte le lezioni e i laboratori sono incoraggiati a partecipare almeno ad alcune di esse e a svolgere tutti gli esercizi svolti durante le lezioni e i laboratori.</p>
<b>Obiettivi formativi specifici e risultati di apprendimento attesi</b>	<p>Il corso appartiene alla tipologia "attività formative di base - informatica di base".</p> <p>Seguendo questo corso, gli studenti saranno in grado di formulare problemi algoritmici e di riconoscere i problemi algoritmici alla base di un'applicazione. Acquisiranno inoltre una conoscenza approfondita delle strutture dati standard e delle corrispondenti tecniche algoritmiche per risolvere tali problemi.</p> <p>Riconosceranno come alcuni approcci algoritmici dipendano dalla scelta di una struttura dati adeguata e viceversa. Inoltre, gli studenti impareranno ad analizzare se un algoritmo è corretto e di quali risorse temporali e spaziali ha bisogno. Infine, gli studenti impareranno a confrontare diversi algoritmi in base alla loro idoneità per una determinata applicazione.</p> <p>Conoscenza e comprensione:</p> <ul style="list-style-type: none"> <li>- D1.3 - Conoscere i principi di base della programmazione.</li> <li>- D1.6 - Conoscere le strutture dati più importanti e il loro utilizzo nei linguaggi di programmazione.</li> </ul>

	<p>Applicazione della conoscenza e della comprensione:</p> <ul style="list-style-type: none"> <li>- D2.2 - Capacità di risolvere problemi algoritmici utilizzando metodi di programmazione.</li> </ul> <p>Capacità di apprendimento</p> <ul style="list-style-type: none"> <li>- D5.1 - Capacità di apprendimento per intraprendere ulteriori studi con un alto grado di autonomia.</li> </ul>
<b>Obiettivi formativi specifici e risultati di apprendimento attesi (ulteriori info.)</b>	<p>Il corso appartiene alla tipologia "attività formative di base - informatica di base".</p> <p>Seguendo questo corso, gli studenti saranno in grado di formulare problemi algoritmici e di riconoscere i problemi algoritmici alla base di un'applicazione. Acquisiranno inoltre una conoscenza approfondita delle strutture dati standard e delle corrispondenti tecniche algoritmiche per risolvere tali problemi.</p> <p>Riconosceranno come alcuni approcci algoritmici dipendano dalla scelta di una struttura dati adeguata e viceversa. Inoltre, gli studenti impareranno ad analizzare se un algoritmo è corretto e di quali risorse temporali e spaziali ha bisogno. Infine, gli studenti impareranno a confrontare diversi algoritmi in base alla loro idoneità per una determinata applicazione.</p>
<b>Modalità di esame</b>	<p>La valutazione si basa su un esame finale scritto.</p> <p>L'esame scritto consiste in domande di verifica delle conoscenze, domande che valutano la capacità di applicare le conoscenze acquisite nel corso ed esercizi.</p>
<b>Criteri di valutazione</b>	<p>Non ci sono requisiti per partecipare all'esame finale.</p> <p>Nell'esame scritto, gli studenti devono applicare le tecniche insegnate nel corso in un contesto definito e devono sviluppare algoritmi per nuovi problemi. Gli algoritmi sviluppati devono essere analizzati in termini di correttezza ed efficienza.</p> <p>Le risposte sono valutate in base alla loro correttezza, all'adeguatezza degli algoritmi sviluppati e alla validità e chiarezza dell'analisi.</p>
<b>Bibliografia obbligatoria</b>	<p>Introduction to Algorithms, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest e Clifford Stein (CLRS), seconda o terza edizione</p> <p>Biblioteca Universitaria: ST 134 C811</p>
<b>Bibliografia facoltativa</b>	<p>Algorithms and Data Structures - The Basic Toolbox, K. Mehlhorn e P. Sanders, scaricabile gratuitamente da</p>

	<a href="http://www.mpi-inf.mpg.de/~mehlhorn/ftp/Mehlhorn-Sanders-Toolbox.pdf">http://www.mpi-inf.mpg.de/~mehlhorn/ftp/Mehlhorn-Sanders-Toolbox.pdf</a>
<b>Altre informazioni</b>	Bibliotecario di soggetto: David Gebhardi, David.Gebhardi@unibz.it Software utilizzato: Compilatore e debugger Java/C
<b>Obiettivi di Sviluppo Sostenibile (SDGs)</b>	Istruzione di qualità