

Principali informazioni sull'insegnamento	
Titolo insegnamento	Ingegneria della conoscenza e Sistemi Esperti
Corso di studio	Informatica L.T.
Crediti formativi	6
Denominazione inglese	Knowledge Engineering and Expert Systems
Obbligo di frequenza	-
Lingua di erogazione	Italiano

Docente responsabile	Nome Cognome	Indirizzo Mail
	Floriana Esposito	floriana.esposito@uniba.it

Dettaglio credi formativi	Ambito disciplinare	SSD	Crediti
	Informatica	Ing-Inf/05	6

Modalità di erogazione	
Periodo di erogazione	Semestre primo
Anno di corso	terzo
Modalità di erogazione	Lezioni frontali 22 lezioni frontali per 47 ore (32 ore lezioni teoriche e 15 esercitazioni guidate) 25 ore di progetto individuale

Organizzazione della didattica	
Ore totali	47+103 = 150 corrispondenti a 6 CFU
Ore di corso	47
Ore di studio individuale	25 di progetto individuale 68 corrispondenti alle lezioni teoriche 10 corrispondenti alle esercitazioni guidate TOTALE 103 ore

Calendario	
	25 settembre 2017
Fine attività didattiche	12 gennaio 2018

Syllabus	
Prerequisiti	Conoscenza dei paradigmi di programmazione, conoscenza delle strutture dati dinamiche e delle nozioni di algoritmica (tecniche selettive e tecniche generative). Competenze specifiche e abilità su programmazione orientata ad oggetti
Risultati di apprendimento previsti (declinare rispetto ai Descrittori di Dublino) (si raccomanda che siano coerenti con i risultati di apprendimento del CdS, compreso i risultati di apprendimento trasversali)	<ul style="list-style-type: none"> <i>Conoscenza e capacità di comprensione</i> <p>Lo studente avrà conoscenza dei fondamenti essenziali della Intelligenza Artificiale, della programmazione euristica e della programmazione a regole. Accanto ai principi della astrazione, alle teorie formali del calcolo attraverso modelli algebrico-matematici, imparerà a risolvere i problemi mediante la ricerca. Vanno considerati gli aspetti essenziali della disciplina che riimangono inalterati a fronte del cambiamento tecnologico, in modo da avere un sistema di riferimento culturale che trascende il tempo e</p>

le circostanze.

Lo studente avrà consapevolezza delle possibilità e dei limiti delle metodologie informatiche basate su conoscenza e sarà in grado di comprendere quali siano i problemi più adatti ad essere affrontati con tali tecniche (software, hardware, e di rete).

- ***Conoscenza e capacità di comprensione applicate***

In particolare lo studente sarà in grado, da un lato, di comprendere i limiti della programmazione imperativa e, dall'altro, di cogliere le opportunità offerte dall'uso di linguaggi interpretati e dalle tecniche di programmazione a regole per lo sviluppo di sistemi dinamici e la implementazione di agenti software autonomi basati su conoscenza. Queste competenze sono trasferite attraverso lezioni teoriche ed esercitazioni pratiche. Inoltre, lo studente, opportunamente seguito, svilupperà autonomamente, o in un gruppo di non più di tre persone, un sistema esperto in un dominio applicativo di sua scelta o assegnato dal docente, realizzato con un ambiente di programmazione interpretato presentato durante le esercitazioni pratiche.

La verifica dell'acquisizione dei concetti teorici e dei metodi può essere effettuata durante l'anno accademico attraverso prove in itinere. La verifica delle abilità pratiche avviene in fase di esame con la discussione del progetto realizzato dallo studente. Chi non avrà inteso avvalersi delle prove in itinere completerà l'esame orale rispondendo a domande teoriche e risolvendo semplici esercizi.

- ***Autonomia di giudizio***

In particolare, gli studenti saranno in grado di dimostrare la capacità di valutare criticamente quanto appreso, formulando un proprio punto di vista ed essendo in grado di sostenerlo nell'ambito di un gruppo di lavoro, operando così in modo efficace come individuo all'interno di una squadra.

L'autonomia di giudizio è acquisita dai discenti attraverso i problemi posti loro a lezione e durante la fase di sviluppo del progetto in autonomia. Tale aspetto è verificato durante gli esami orali e soprattutto in sede di discussione del progetto: si potrà così mettere in chiaro le scelte progettuali implementate ed anche evincere i contributi personali di ogni studente partecipante al gruppo di lavoro.

- ***Abilità comunicative***

Il tipo di professionalità che implica il lavorare in gruppo e l'esigenza di interloquire con i committenti e gli utenti finali allo scopo di comprenderne le esigenze e rappresentare loro efficacemente i ritorni delle scelte progettuali fatte, impone l'identificazione e l'acquisizione di abilità che vanno oltre le competenze tecniche. Queste abilità sono assicurate in fase di sviluppo del progetto sia in autonomia che in gruppo. Durante questa fase la identificazione delle sorgenti di conoscenza, l'utilizzo di tecniche appropriate per acquisire la conoscenza dagli esperti, la discussione col docente e con i colleghi del gruppo di lavoro delle possibili scelte progettuali e delle soluzioni implementative forza alla rappresentazione, alla comunicazione e alla discussione delle proprie idee.

- ***Capacità di apprendere***

Essendo un insegnamento del terzo anno si suppone che gli studenti abbiano già un alto livello di autonomia nell'apprendimento e nell'approccio metodologico, capacità che consente loro di affrontare studi successivi e/o di proseguire il proprio percorso formativo in modo autonomo, tenendosi aggiornati rispetto alla continua evoluzione tecnologica caratteristica della disciplina. Lo studente deve essere in grado di consultare materiale bibliografico tradizionale o reperibile via internet o attraverso piattaforme di e-learning, deve essere capace di sintetizzare quanto scritto in libri di testo e in manuali prevalentemente in Inglese e utilizzarlo in fase di sviluppo di progetto e deve esporre quanto appreso durante la prova orale. Le linee guida all'apprendimento sono date dai contenuti delle slide delle

Contenuti di insegnamento	lezioni e delle esercitazioni tenute dal docente, materiale a disposizione degli studenti. L'esposizione, sia scritta che orale, dell'elaborato finale del progetto e il colloquio orale rappresentano il momento di verifica di tali capacità.
	<p>Obiettivi Formativi</p> <ul style="list-style-type: none"> ❑ Conoscenza dei principi della programmazione euristica; soluzione dei problemi mediante ricerca; conoscenza del paradigma di programmazione a regole. <p>Professionalizzanti</p> <ul style="list-style-type: none"> ❑ Capacità di utilizzare ambienti a regole per lo sviluppo di sistemi basati su conoscenza e di integrare paradigmi di programmazione alternativi per risolvere problemi di aiuto alle decisioni, classificazione, diagnosi. Abilità a progettare sistemi di programmazione euristica. <p>Contenuti</p> <ul style="list-style-type: none"> ❑ DALLA INGEGNERIA DELL'INFORMAZIONE ALLA INGEGNERIA DELLA CONOSCENZA ❑ LA PROGRAMMAZIONE EURISTICA ❑ IL MODELLO DI CALCOLO: (<i>i Production Systems</i>) ❑ AGENTI BASATI SU CONOSCENZA ❑ LA PROGRAMMAZIONE A REGOLE ❑ LA INGEGNERIZZAZIONE DEI SISTEMI BASATI SU CONOSCENZA: DALLA ACQUISIZIONE ALLA RAPPRESENTAZIONE DELLA CONOSCENZA ❑ LA QUALITÀ DELLA CONOSCENZA: RAGIONARE CON CONOSCENZA INCERTA

Programma	
Testi di riferimento	
	<p>Di base: Elaine Rich, Kevin Knight, Shivashankar B. Nair, "Artificial Intelligence" third edition Mc Graw Hill, 2008 S. J. Russell, P. Norvig, "Intelligenza Artificiale" Prentice Hall, Vol. I, 2005</p> <p>Riferimenti: Nils J. Nilsson "Principles of Artificial Intelligence" Morgan Kaufmann, 1980 Peter Jackson "Introduction to Expert Systems" Addison Wesley, 1998 Mark Stefik "Knowledge Systems", Morgan Kaufmann, 1995 Keith Darlington, "The essence of expert systems", Prentice Hall, 2000</p> <p>Lucidi lezioni Programmazione a regole: Manuali dei sistemi</p>
Metodi didattici	<p>Lezioni frontali teoriche Esercitazioni guidate Sviluppo di un progetto</p>
Metodi di valutazione (indicare almeno la tipologia scritto, orale, altro)	<p>Esame orale: domande sulla teoria e discussione del progetto consegnato. Il progetto va consegnato su supporto stabile insieme alla documentazione che deve rispondere alle domande</p> <ol style="list-style-type: none"> 1. è utile? 2. quale è il dominio e il compito del sistema? 3. che esigenza c'è dell'interprete e della interlocuzione in

	<p>tempo reale?</p> <ol style="list-style-type: none"> 4. chi sono gli utenti? 5. il sistema è in grado di riconoscerli? 6. si potrebbe realizzare il sistema in un ambiente di programmazione tradizionale? 7. la conoscenza è stabile, è incerta? <p>La documentazione del progetto deve inoltre comprendere l'elenco delle regole (espresse in linguaggio naturale) e due /tre esempi di sessioni di lavoro. Il progetto viene discusso in sede di esame orale.</p>
<p>Criteri di valutazione (per ogni risultato di apprendimento atteso su indicato, descrivere cosa ci si aspetta lo studente conosca o sia in grado di fare e a quale livello al fine di dimostrare che un risultato di apprendimento è stato raggiunto e a quale livello)</p>	<p>Lo studente deve essere in grado di rispondere alle domande sulla teoria dimostrando di aver compreso le caratteristiche del modello di calcolo a produzioni rispetto al modello di Turing. Per quanto riguarda la programmazione euristica, deve essere in grado di descrivere gli algoritmi di ricerca generalizzata, ricerca informata, di rappresentare l'uso delle funzioni euristiche, di valutare le prestazioni degli algoritmi di ricerca euristici, di descrivere la ricerca a due avversari e di saper applicare quanto detto in semplici esercizi. Deve essere in grado di descrivere la realizzazione del modello a produzioni nella famiglia degli ambienti di calcolo OPS e le scelte implementative dell'algoritmo RETE. Deve essere in grado di rappresentare le particolarità progettuali di sistemi esperti con compiti di classificazione, diagnosi e pianificazione. Deve saper adeguatamente descrivere i metodi per il trattamento di conoscenza in condizioni di incertezza (mediante fattori di certezza e regole fuzzy).</p> <p>Per quanto riguarda il progetto, sviluppato in ambiente CLIPS, lo studente deve dimostrarne il funzionamento, la qualità della conoscenza inserita, la capacità di modellare adeguatamente lo scenario d'uso e l'utente, la capacità di ritrattare e riavviare un nuovo ragionamento con l'aggiornamento e l'uso appropriato dei fatti immessi in agenda, la possibilità di integrare, in esecuzione, CLIPS con ambienti di programmazione imperativa.</p>
<p>Altro</p>	