

<b>STRUTTURA</b>	Scuola Politecnica - DICGIM
<b>ANNO ACCADEMICO</b>	2014-15
<b>CORSO DI LAUREA MAGISTRALE</b>	Ingegneria Informatica
<b>INSEGNAMENTO</b>	<b>Architetture avanzate dei calcolatori (C.I.)</b>
<b>TIPO DI ATTIVITÀ</b>	Caratterizzante
<b>AMBITO DISCIPLINARE</b>	Ingegneria informatica
<b>CODICE INSEGNAMENTO</b>	16975
<b>ARTICOLAZIONE IN MODULI</b>	SI
<b>NUMERO MODULI</b>	2
<b>SETTORI SCIENTIFICO DISCIPLINARI</b>	ING-INF/05
<b>DOCENTE RESPONSABILE</b>	Giuseppe Lo Re Professore associato Università degli Studi di Palermo giuseppe.lore@unipa.it
<b>ALTRO DOCENTE DI RIFERIMENTO</b>	Daniele Peri Ricercatore Università degli Studi di Palermo daniele.peri@unipa.it
<b>CFU</b>	12
<b>NUMERO DI ORE RISERVATE ALLO STUDIO PERSONALE</b>	192
<b>NUMERO DI ORE RISERVATE ALLE ATTIVITÀ DIDATTICHE ASSISTITE</b>	108
<b>PROPEDEUTICITÀ</b>	NO
<b>ANNO DI CORSO</b>	Primo
<b>SEDE DI SVOLGIMENTO DELLE LEZIONI</b>	Consultare il sito <a href="http://politecnica.unipa.it">politecnica.unipa.it</a>
<b>ORGANIZZAZIONE DELLA DIDATTICA</b>	Lezioni frontali Esercitazioni in aula e nelle aule informatiche
<b>MODALITÀ DI FREQUENZA</b>	Facoltativa
<b>METODI DI VALUTAZIONE</b>	Prova scritta e prova orale
<b>TIPO DI VALUTAZIONE</b>	Voto in trentesimi
<b>PERIODO DELLE LEZIONI</b>	Consultare il sito <a href="http://politecnica.unipa.it">politecnica.unipa.it</a>
<b>CALENDARIO DELLE ATTIVITÀ DIDATTICHE</b>	Consultare il sito <a href="http://politecnica.unipa.it">politecnica.unipa.it</a>
<b>ORARIO DI RICEVIMENTO DEGLI STUDENTI</b>	Lunedì ore 14.00-16.00

#### **RISULTATI DI APPRENDIMENTO ATTESI**

##### **Conoscenza e capacità di comprensione**

Lo studente conoscerà l'evoluzione storica e lo stato dell'arte dei sistemi di elaborazione dal punto di vista strutturale. Conoscerà quindi i principi di funzionamento delle principali architetture dei calcolatori e dei loro componenti. Avrà conoscenza approfondita della programmazione a basso livello, di sistema ed embedded e dei relativi strumenti di sviluppo. Lo studente saprà inoltre valutare le caratteristiche e l'innovatività delle soluzioni architetture disponibili.

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

Lo studente sarà in grado di applicare le conoscenze acquisite prospettando possibili soluzioni a problemi di programmazione di sistema, a basso livello ed embedded, e saprà affrontarne l'implementazione utilizzando gli strumenti e ambienti di sviluppo più appropriati.

### **Autonomia di giudizio**

Lo studente sarà in grado di affrontare in autonomia l'analisi, la progettazione e l'implementazione di soluzioni per problemi di programmazione a basso livello, di sistema ed embedded e sarà in grado di valutarne la qualità in termini di semplicità, versatilità, efficienza, riutilizzabilità ed economicità.

### **Abilità comunicative**

Lo studente sarà in grado di esporre, efficacemente e con proprietà di linguaggio, i principi strutturali e di funzionamento dei calcolatori. Saprà inoltre presentare analisi e soluzioni di problemi affrontabili con la programmazione a basso livello, di sistema ed embedded.

### **Capacità d'apprendimento**

Lo studente sarà in grado di affrontare in maniera autonoma lo studio delle architetture dei calcolatori, con particolare riferimento a quanto disponibile in commercio e nella letteratura scientifica. Saprà affrontare problemi di programmazione a basso livello, di sistema ed embedded, individuando e integrando soluzioni parziali già disponibili, sia formalizzate sia implementate. Sarà in grado di approfondire in autonomia la conoscenza di strumenti di progettazione e di sviluppo. Sarà inoltre in grado di integrare le conoscenze acquisite negli altri insegnamenti con quelle del corso.

### **OBIETTIVI FORMATIVI**

Al termine del corso lo studente conoscerà i concetti avanzati necessari alla comprensione della struttura dei calcolatori elettronici **e la loro evoluzione storica**. Conoscerà la struttura dei calcolatori e dei loro componenti secondo i principali modelli architettureali. Avrà conoscenza approfondita della programmazione di sistema, a basso livello, ed embedded e dei relativi strumenti di progettazione e sviluppo. Lo studente sarà in grado di valutare, analizzare, comunicare e implementare le possibili soluzioni a problemi applicativi di programmazione di sistema, a basso livello, ed embedded.

<b>MODULO 1</b>	
<b>ORE FRONTALI</b>	<b>LEZIONI FRONTALI</b>
27	<b>Storia dell'informatica moderna.</b> L'avvento dell'elaborazione dei dati nelle aziende. L'introduzione del software. Il Mainframe IBM. Dal Mainframe al Minicomputer. L'invenzione del Chip e il suo impatto. Il Personal Computer. Workstation, UNIX, e la Rete.
<b>MODULO 2</b>	
<b>ORE FRONTALI</b>	<b>LEZIONI FRONTALI</b>
33	<p><b>Struttura dei calcolatori.</b></p> <p><b>Processori.</b> Struttura delle CPU. Esecuzione delle istruzioni. RISC e CISC. Parallelismo a livello di istruzione. Parallelismo a livello di processore.</p> <p><b>Memoria.</b> Indirizzamento. Ordinamento dei byte. Memoria cache. Registri.</p> <p><b>Bus.</b> Tipi di bus e loro gestione. Operazioni sui bus. Bus PCI. Bus USB. I/O su bus e mappato in memoria.</p> <p><b>Microarchitetture.</b> Microistruzioni. Ottimizzazione dell'esecuzione mediante memoria cache, predizione dei salti, esecuzione Out-of-order e ridenominazione dei registri. Esecuzione speculativa.</p> <p><b>Architettura a livello di istruzione.</b> Tipi di dato. Formati e tipi delle istruzioni. Modi di indirizzamento. Flusso di controllo. Interrupt.</p> <p><b>Supporto al sistema operativo.</b> Memoria virtuale. Virtualizzazione hardware. Istruzioni di I/O. Istruzioni per l'elaborazione parallela.</p> <p><b>Architetture parallele.</b> Parallelismo a livello di chip. Coprocessori. Multiprocessori a memoria condivisa.</p> <p><b>Esempi di architetture di processori.</b> Intel x86, ARM e Atmel AVR.</p>
21	<p><b>Programmazione a basso livello, di sistema ed embedded.</b></p> <p>Linguaggio macchina e di assemblatore. Traduzione delle istruzioni. Macro. Traduzione dal linguaggio di assemblatore. Tavole dei simboli. Collegamento e caricamento. Moduli oggetto. Collegamento statico e dinamico. Programmazione di microcontrollori. Programmazione in Forth.</p>
	<b>ESERCITAZIONI</b>
27	Programmazione a basso livello, di sistema ed embedded.
<b>TESTI CONSIGLIATI</b>	<p>A. S. Tanenbaum, T. Austin, "Structured computer organization. 6th ed.", Pearson</p> <p>D. A. Patterson, J. L. Hennessy, "Computer Organization and Design", Morgan Kaufmann</p> <p>L. Brodie, "Starting Forth", disponibile on-line</p> <p>L. Brodie, "Thinking Forth", disponibile on-line</p>