



# UNIVERSITÀ DEGLI STUDI DI PALERMO

<b>DEPARTMENT</b>	Ingegneria
<b>ACADEMIC YEAR</b>	2015/2016
<b>MASTER'S DEGREE (MSC)</b>	COMPUTER ENGINEERING
<b>SUBJECT</b>	ADVANCED COMPUTER ARCHITECTURES
<b>TYPE OF EDUCATIONAL ACTIVITY</b>	B
<b>AMBIT</b>	50369-Ingegneria informatica
<b>CODE</b>	01475
<b>SCIENTIFIC SECTOR(S)</b>	ING-INF/05
<b>HEAD PROFESSOR(S)</b>	GENTILE ANTONIO      Professore Associato      Univ. di PALERMO
<b>OTHER PROFESSOR(S)</b>	
<b>CREDITS</b>	6
<b>INDIVIDUAL STUDY (Hrs)</b>	96
<b>COURSE ACTIVITY (Hrs)</b>	54
<b>PROPAEDEUTICAL SUBJECTS</b>	
<b>MUTUALIZATION</b>	
<b>YEAR</b>	1
<b>TERM (SEMESTER)</b>	1° semester
<b>ATTENDANCE</b>	Not mandatory
<b>EVALUATION</b>	Out of 30
<b>TEACHER OFFICE HOURS</b>	<b>GENTILE ANTONIO</b> Friday      10:00    12:00    Studio del docente presso DINFO, Edificio 6, III pianoDietro prenotazione per email/sito o telefono: 091-238.62603

DOCENTE: Prof. ANTONIO GENTILE

<b>PREREQUISITES</b>	
<b>LEARNING OUTCOMES</b>	<p>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 e dei relativi strumenti di sviluppo. Lo studente saprà inoltre valutare le caratteristiche e l'innovatività delle soluzioni architetture disponibili.</p> <p>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, e saprà affrontarne l'implementazione utilizzando gli strumenti e ambienti di sviluppo più appropriati.</p> <p>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 e sarà in grado di valutarne la qualità in termini di semplicità, versatilità, efficienza, riutilizzabilità ed economicità.</p> <p>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 e di sistema.</p> <p>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 e di sistema, 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.</p>
<b>ASSESSMENT METHODS</b>	Prova scritta. Discussione su un elaborato di progetto concordato.
<b>EDUCATIONAL OBJECTIVES</b>	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 architetture. Avrà conoscenza approfondita della programmazione di sistema, a basso livello, 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 e a basso livello.
<b>TEACHING METHODS</b>	Lezioni frontali Esercitazioni in aula
<b>SUGGESTED BIBLIOGRAPHY</b>	A. S. Tanenbaum, T. Austin, "Structured computer organization. 6th ed.", Pearson D. A. Patterson, J. L. Hennessy, "Computer Organization and Design", Morgan Kaufmann Note fornite dal docente attraverso il blog del corso

## SYLLABUS

Hrs	Frontal teaching
3	Storia dell'informatica moderna: L'avvento dell'elaborazione dei dati nelle aziende.
3	Storia dell'informatica moderna: L'introduzione del software.
3	Storia dell'informatica moderna: Il Mainframe IBM.
3	Storia dell'informatica moderna: Dal Mainframe al Minicomputer.
3	Storia dell'informatica moderna: L'invenzione del Chip e il suo impatto.
3	Storia dell'informatica moderna: Il Personal Computer.
3	Storia dell'informatica moderna: Workstation, UNIX, e la Rete.
4	Processori. Struttura delle CPU. Esecuzione delle istruzioni. RISC e CISC. Parallelismo a livello di istruzione. Parallelismo a livello di processore.
4	Memoria. Indirizzamento. Ordinamento dei byte. Memoria cache. Registri.
4	Bus. Tipi di bus e loro gestione. Operazioni sui bus. Bus PCI. Bus USB. I/O su bus e mappato in memoria.
4	Microarchitetture. Microistruzioni. Ottimizzazione dell'esecuzione mediante memoria cache, predizione dei salti, esecuzione Out-of-order e ridenominazione dei registri. Esecuzione speculativa.

## SYLLABUS

Hrs	Frontal teaching
4	Architettura a livello di istruzione. Tipi di dato. Formati e tipi delle istruzioni. Modi di indirizzamento. Flusso di controllo. Interrupt.
4	Supporto al sistema operativo. Memoria virtuale. Virtualizzazione hardware. Istruzioni di I/O. Istruzioni per l'elaborazione parallela.
4	Architetture parallele. Parallelismo a livello di chip. Coprocessori. Multiprocessori a memoria condivisa. Esempi di architetture di processori.
3	Programmazione a basso livello e di sistema: Linguaggio macchina e di assembler. Traduzione delle istruzioni. Macro.
3	Programmazione a basso livello, di sistema ed embedded. Traduzione dal linguaggio di assembler. Tavole dei simboli. Collegamento e caricamento.
3	Programmazione a basso livello, di sistema ed embedded: Moduli oggetto. Collegamento statico e dinamico.
2	Programmazione a basso livello, di sistema ed embedded: Programmazione di microcontrollori.