



SCHEMA DELL'INSEGNAMENTO (SI)

"ADVANCED COMPUTER PROGRAMMING"

SSD ING-INF/05

DENOMINAZIONE DEL CORSO DI STUDIO: CORSO DI LAUREA IN INGEGNERIA INFORMATICA

ANNO ACCADEMICO: 2023-2024

INFORMAZIONI GENERALI - DOCENTE

DOCENTE: RAFFAELE DELLA CORTE

TELEFONO: 0817683820

EMAIL: raffaele.dellacorte2@unina.it

SI VEDA SITO WEB DEL CORSO DI STUDI

INFORMAZIONI GENERALI - ATTIVITÀ

INSEGNAMENTO INTEGRATO (EVENTUALE): N.A.

MODULO (EVENTUALE): N.A.

CANALE (EVENTUALE): N.A.

ANNO DI CORSO (I, II, III): III

SEMESTRE (I, II): II

CFU: 9



INSEGNAMENTI PROPEDEUTICI (se previsti dal Regolamento del CdS)

Programmazione.

EVENTUALI PREREQUISITI

Nessuno.

OBIETTIVI FORMATIVI

Il corso ha l'obiettivo di fornire conoscenze e competenze di programmazione avanzata in ambito concorrente e distribuito, introducendo gli strumenti per la programmazione ed il debugging di applicazioni multithreading e su rete in linguaggio Java e Python, e fornendo le basi del concetto di middleware e delle diverse soluzioni adottate in ambito industriale, soffermandosi principalmente sul modello orientato ai messaggi e sul modello a servizi, con applicazioni su tecnologie reali. Il corso introduce inoltre gli strumenti per la programmazione di web-application, sia front-end che back-end.

RISULTATI DI APPRENDIMENTO ATTESI (DESCRITTORI DI DUBLINO)

Conoscenza e capacità di comprensione

Lo studente deve dimostrare di conoscere e saper comprendere i problemi di programmazione concorrente e distribuita, principalmente nell'ambito di applicazioni in linguaggio Java e Python, nonché le caratteristiche delle differenti tecnologie middleware. Il percorso formativo intende fornire agli studenti le conoscenze e gli strumenti metodologici, teorici e pratici necessari per riconoscere, analizzare e risolvere problemi legati allo sviluppo di applicativi multithreading e su rete, consentendo agli studenti di padroneggiare lo sviluppo di progetti software avanzati in Java e Python.

Capacità di applicare conoscenza e comprensione

Lo studente deve dimostrare di essere in grado di risolvere problemi concernenti la programmazione concorrente e distribuita, utilizzando le competenze metodologiche, teoriche e pratiche di programmazione avanzata presentate al corso per la realizzazione di progetti software, multithreading e su rete, in linguaggio Java e Python.

PROGRAMMA-SYLLABUS

Programmazione concorrente e su rete in Java.

Richiami sul linguaggio Java e classi contenitore. Programmazione concorrente in Java. Threads in Java, stati di un thread, pool di threads. Sincronizzazione in Java. Monitor Java e il package `java.util.concurrent` di Java 1.5. Build automation tool edebugging in Java. Programmazione generica, reflection ed annotazioni in Java.

Programmazione su rete in Java. Il package `java.net`. Socket TCP in Java: classi `Socket` e `ServerSocket`. Socket UDP in Java: classi `DatagramSocket` e `DatagramPacket`. Server multithread. Astrazione di oggetto remoto. Proxy-Skeleton.

Modelli di middleware.

Definizione e proprietà del livello middleware. Enterprise Application Integration (EAI). Chiamata di procedura remota (RPC), scambio di messaggi (MOM), elaborazione transazionale (TP), spazio delle tuple (TS), accesso a dati remoti (RDA), oggetti distribuiti (DOM), modello a componenti (CM), web services, microservizi.



Modello a scambio di messaggi.

Specifica Java Message Service (JMS), client e provider. Comunicazione point-to-point e publish-subscribe. Modello di programmazione JMS. Messaggi JMS ed aspetti avanzati.

Modello a servizi ed implementazione web-app application.

Cenni a SOAP e ai servizi RPC. RESTful Web Services, risorsa e Uniform Resource Identifier (URI). Servizi RESTful e metodi HTTP. Implementazione di RESTful Web Services con framework Java ed implementazione di web-app con HTML, Javascripte framework per lo sviluppo front-end.

Il linguaggio Python.

Tipi di dato, costrutti di controllo, passaggio parametri, data collection, file, funzioni, moduli e debugging in Python. Programmazione object-oriented in Python: classi, oggetti, ereditarietà, polimorfismo, classi astratte. Programmazione concorrente e su rete in Python: multithreading, sincronizzazione, socket. Esempi di integrazione multi-linguaggio. DataScience in Python.

MATERIALE DIDATTICO

- trasparenze delle lezioni del corso.
- Libri di testo:
 - B. Eckel *"Thinking in Java"*.
 - L. H. Etzkorn *"Introduction to Middleware - Web Services, Object Components, and Cloud Computing"*.
 - Semmy Purewal *"Learning Web App Development"*.
 - Craig Walls *"Spring in Action"*.
 - Allen B. Downey *"Think Python"*.
 - Mark Lutz *"Programming Python"*.
- Materiale esercitativo.
- Risorse disponibili in rete.

MODALITÀ DI SVOLGIMENTO DELL'INSEGNAMENTO

Lezioni frontali ed esercitazioni guidate svolte in aula.

VERIFICA DI APPRENDIMENTO E CRITERI DI VALUTAZIONE

a) Modalità di esame:

L'esame si articola in prova	
scritta e orale	X
solo scritta	
solo orale	
discussione di elaborato progettuale	
altro	Prova al calcolatore

In caso di prova scritta i quesiti sono	A risposta multipla	
	A risposta libera	
	Esercizi numerici	

- b) Modalità di valutazione:** L'esito della prova scritta è vincolante ai fini dell'accesso alla prova orale. Si prevede inoltre la possibilità di pianificare prove intercorso eventualmente a sostituzione della prova scritta.