

*University of Pisa*  
*and*  
*Scuola Superiore Sant'Anna*

*Master Degree Program in Computer Science and Networking*

**Major (mandatory) Courses**

**academic year 2013-14**

**SYLLABUS**

## *Algorithm Engineering (9 CFU)*

### **Algorithm Engineering**

In this course we will study, design and analyze (theoretically and experimentally) advanced algorithms and data structures for the efficient solution of combinatorial problems involving all basic data types, such as integers, strings, (geometric) points, trees and graphs. These algorithmic tools will be designed and analyzed in several models of computation— such as RAM, 2-level memory, cache-oblivious, streaming— in order to take into account the architectural features and the memory hierarchy of modern PCs.

Every lecture will follow a problem-driven approach that starts from a real software-design problem, abstracts it in a combinatorial way (suitable for an algorithmic investigation), and then introduces algorithmic solutions aimed at minimizing the use of some computational resources like time, space, communication, I/O, etc. Some of these solutions will be discussed at an experimental level, in order to introduce proper engineering and tuning tools for algorithmic development.

#### *Syllabus*

- 1) RAM model
  - a. Data compression
  - b. Data processing: randomized, adaptive, self-adjusting.
  - c. Data indexing and searching: strings, geometric points, trees and graphs
- 2) 2-level memory model
  - a. Definition and properties
  - b. Data sorting and permuting
  - c. Data indexing and searching: strings and multi-dimensional data
  - d. Data sketching: bloom filters and count-min sketch
- 3) Cache-oblivious model
  - a. Definition and properties
  - b. Matrix multiplication
  - c. VEB layout
  - d. Tree mapping

Written and oral exam.

*Fondamenti di Segnali, Sistemi e Reti (12 CFU)***Fundamentals of Signals, Systems and Networks**

This course introduces the fundamentals of signal theory, of stochastic processes, the fundamentals of queuing theory, some basic elements of electromagnetism and some calculus. The course will also cover the main network architectures for access, metropolitan and core segments.

***Syllabus***

- 1) Signal Theory, basic calculus
  - a. Finite energy and finite power discrete and continuous signals
  - b. Periodic signals
  - c. Time invariant linear systems
  - d. Description of signals and systems in the frequency domain
  - e. Advanced calculus
- 2) Processi stocastici e teoria delle code
  - a. General concepts
  - b. Probability and random variables
  - c. Stochastic processes
  - d. Markov chain and process
  - e. Elements of queuing theory
- 3) Design of networks
  - a. Network hierarchy
  - b. Access segment
  - c. Metropolitan segment
  - d. Core segment
  - e. Future architectures

Exam consists in a written test concerning course concepts and a possible discussion of a project assigned to the student.

## *Gestione e configurazione delle reti (9 CFU)*

### **Network Configuration and Management**

This course aims at providing students with a guided and critical overview of the communications network protocol evolution. The course will mainly focus on the TCP/IP architecture of the current Internet and on the Ethernet protocol. Finally the most recently introduced standards will be presented. Protocols (e.g., MPLS) and algorithms for traffic engineering and resilience will be introduced. Elements of event driven simulation will be provided. The concept presented in class will become topic of lab experiments.

#### ***Syllabus***

##### 1) Communications Networks

- Data Transmission fundamentals
  - Digital vs Analog Transmission
  - Analog to Digital Conversion
  - Physical Media
  - Connection-oriented vs connectionless communications
  - Packet switching
- Review of ISO/OSI and TCP/IP protocol architecture
- MAC and Logical Link Control
  - Automatic Repeat reQuest (ARQ)
  - Pipelining
- Ethernet
  - CSMA/CD
  - Repeaters, Bridges, Switches
  - Switched Ethernet
    - 1GbE, 10GbE, beyond 10GbE
  - VLAN
- IP Protocols
  - Classes
  - Subnetting
  - Supernetting
  - Routing basics
    - OSPF
  - Quality of service
    - IntServ
    - DiffServ
- MPLS protocols
  - OSPF-TE
  - RSVP-TE
- Optical Networks
  - SONET/SDH
  - Wavelength Division Multiplexing (WDM)
- Network resilience and availability
  - Resilience in IP, MPLS and Optical Networks
  - Introduction to availability theory
- Introduction to event driver simulation

## 2) Lab of Traffic Engineering

- Network nodes: routers and OADM architectures
- Juniper routers: basic configuration
- OSPF
- IP Routing
- OSPF-TE
- MPLS Label Switch Path
- MPLS Path Recovery
- MPLS Local Recovery
- Project

Exam consists in written test concerning course concepts and a possible discussion of a project assigned to the student.

## *High Performance Computing (9 CFU)*

### **High Performance Computing**

This course deals with two interrelated issues in high-performance computing:

1. fundamental concepts and techniques in parallel computation structuring and design, including parallelization methodologies and paradigms, parallel programming models, their implementation, and related cost models;
2. architectures of high-performance computing systems, including shared memory multiprocessors, distributed memory multicomputers, clusters, and others.

Both issues are studied in terms of structural model, static and dynamic support to computation and programming models, performance evaluation, capability for building complex and heterogeneous applications and/or enabling platforms, also through examples of application cases. Technological features and trends are studied, in particular multi-/many-core technology and high-performance networks.

An initial part is dedicated to review basic concepts and techniques in structured computer architecture, in order to render the different backgrounds of students as uniform as possible.

#### ***Syllabus***

1. A structured computer architecture primer: review of level structuring, processing modules, firmware architecture, assembler machine, memory hierarchies and caching, process level and interprocess communication
2. Methodology for structuring and programming high-performance parallel applications, basic cost models: metrics, elements of queueing theory and queueing networks, load balancing, static and dynamic optimizations
3. Parallel paradigms: stream-parallel (pipeline, data-flow, farm, divide and conquer, functional partitioning), data-parallel (map, fixed and variable stencils, reduce, prefix), and their compositions
4. Run-time supports of parallel programs and their optimization
5. Shared memory multiprocessors: SMP and NUMA, cost models; interconnection networks and their evaluation: indirect and multistage networks, direct and cube networks, fat tree, on-chip networks
6. Distributed memory architectures: multicomputers, clusters, distributed heterogeneous platforms, high-performance communication networks
7. Advanced research and/or technological issues: multi-/many-core, multithreading simd/vectorization/gpu, pervasive high-performance computing.

Exam consists in a written and an oral part.

## *Ingegneria del teletraffico (9 CFU)*

### **Teletraffic Engineering**

The course gives the fundamentals concepts related to Teletraffic Theory and its application to network engineering. The aim of the course is to give the students the capacity of building up and analyse their own abstraction of basic functions related to telecommunication networks or discrete state stochastic systems in general. Transient and Steady-state analysis of Discrete and Continuous Time Markov processes are introduced. Fundamentals concept related to Queueing theory and their application to circuit and packet switching networks are presented. The analysis of fundamental performance indexes is carried out, when necessary, by means of the transforms theory (e.g. Laplace, Zeta). The fundamental theorems related to the tractability of open and closed Queueing Networks are also presented. The classroom and laboratory (matlab) exercise are aimed to give the student the ability to carry out the solution of basic cases by proper analytical or numerical methods.

- 1) Discrete state Markov processes
  - a. Discrete State, Discrete Time Markov Processes (Markov chains)
  - b. Discrete State, Continuous Time Markov Processes
- 2) Point Processes
  - a. Pure Birth and Pure Death processes
  - b. Discrete time and continuous time Bernoulli processes
  - c. Poisson process
- 3) Birth and Death Processes
  - a. Ergodic conditions
  - b. First and second order momentum
- 4) Basics on teletraffic analysis
  - a. Stochastic models
  - b. Deterministic models
  - c. Non stationary behavior. TCBH, ADPH definitions
- 5) Markovian queues
  - a. Kendall notation; Geo/Geo/1, M/M/Ns, M/M/Ns/0, M/M/1/Nw;
  - b. Erlang B Formula, Erlang C Formula, Engset formula.
  - c. Problems and solutions related to the evaluation of Erlang B e Erlang C formulas. MATLAB functions for the evaluation of loss probability in M/M/1/Ns and M/M/Ns/Nw queues. MATLAB solutions for M/Cox2/1/Nw, M/H2/1/Nw and M/E2/1/Nw queues.
  - d. Matrix-Geometric approach for the solution of Markov chains described by block Hessenberg matrices.
  - e. Matrix-Geometric approach for the analysis of the M/Cox2/1 queue.
- 6) Non Markovian queues:
  - a. A simple non markovian queue: the M/G/1 queue;
  - b. The embedded Markov chain; steady state analysis of the M/G/1 queue.
  - c. Multiple user classes and priorities in M/G/1 queues
- 7) Queueing networks:
  - a. Open and Closed Markovian Queueing Networks.
  - b. Burke Theorem. Jackson Theorem.
  - c. Gordon-Newell Theorem.
  - d. Convolution Algorithm and Mean Value Analysis for the solution of the Gordon-Newell queueing networks.
  - e. BCMP queueing networks. Performance indexes in markovian open and closed queueing networks and BCMP.
- 8) Numerical tools for the solution of Markov Chains
  - a. Matlab Matfun e Stats Libraries. Erlang k, iperexponential, ipoexponential and Coxn generators

- b. Quantile-quantile plots. Eigenvalues decomposition for the transient analysis of Markov chains.
- c. Direct methods for the evaluations of the steady state in ergodic Markov Chains.

Written test, laboratory test on MATLAB, and Oral Exam.

## *Ingegneria dei servizi software (9 CFU)*

### **Software Service Engineering**

The overall objective of the course is to introduce the main aspects in the design and development of software services. After introducing some of the currently adopted standards for Web services, the course centers on service-oriented architectures and on the techniques for developing applications by discovering, composing and adapting existing services. The use of languages supporting the specification and the execution of business processes via workflows is discussed. The course also describes some of the techniques employed for guaranteeing service properties. Finally, some recent developments and trends in software service engineering are introduced.

#### ***Syllabus***

- Web service core standards
  - messaging
  - service description
  - service registration and discovery
- Service-oriented software development
  - service-oriented architectures
  - service composition
    - service orchestration and choreographies
    - business process modelling and analysis
    - business process execution languages
    - transactions
  - service properties
    - service policies
    - service level agreements
  - service development lifecycle
- Introduction to recent developments and trends in software service engineering

Part of the classes are lab sessions. To get the credits students have to develop a project before undergoing an oral examination.

## *Programmazione Avanzata (9 CFU)*

### **Advanced Programming**

The objectives of this course are:

- a. to provide the students with a deep understanding of how high level programming concepts and metaphors map into executable systems and which are their costs and limitations
- b. to acquaint the students with modern principles, techniques, and best practices of sophisticated software construction
- c. to introduce the students to techniques of programming at higher abstraction levels, in particular generative programming, component programming and web computing
- d. to present state-of-the-art frameworks incorporating these techniques.

This course focuses on the quality issues pertaining to detailed design and coding, such as reliability, performance, adaptability and integrability into larger systems.

#### ***Syllabus***

1. Programming Language Pragmatics
2. Run Time Support and Execution Environments
3. Generic Programming
4. Class Libraries and Frameworks
5. Generative Programming
6. Language Interoperability
7. Component Based Programming
8. Web Services
9. Web and Application Frameworks
10. Scripting Languages

3 CFU on language pragmatics, 3 on programming metaphors, 3 on web programming.

The exam consists in preparing a final term paper, solving a complex programming problem.

*Sistemi distribuiti: paradigmi e modelli (9 CFU)*

**Distributed Systems: Paradigms and Models**

The main course deals with a set of arguments related to the programming models targeting parallel and/or distributed architectures. The common background of these arguments is represented by structured parallel programming models, such as those based on algorithmic skeletons or on parallel design patterns. These programming models are used to discuss:

- the general idea of structured parallel pattern exposed to the application programmer as primitive mechanisms of the parallel programming framework
- the technologies and the paradigms that can be used to efficiently implement structured parallel patterns on different kind of (hardware and software) target architectures including the more common architectures such as network of workstations and multi/many core machines
- a general methodology for the modular design and implementation of parallel applications from a set of predefined building blocks whose parallel pattern, performance features and composition properties are known.

The course includes a “theoretic” part and a lab part. The lab part consists in the presentation of a set of structured parallel programming frameworks as well as of a set of implementation techniques (in classroom) and in the implementation of a simple project (individual) of either a structured parallel application or of a run time support for some structured parallel pattern.

**Syllabus**

1. Structured parallel programming models based on algorithmic skeletons: expressive power, types of skeletons, programming model, composition, semantics, functional and non functional parameters, implementation techniques based on process templates and on macro data flow, performance models and optimization techniques.
2. Implementation of parallel patterns: design and implementation of template for parallel/distributed architectures, techniques for distributed / shared memory architectures, transformation and optimization techniques, management of non functional features in parallel computations.
3. Methodology to support compositional development of parallel applications: definition of the building block set, structured design techniques, monitoring techniques and non functional feature management strategies.

The exam consists in an individual project plus an oral exam.