Admission exam procedure and syllabus

The admission exam has to achieve two main goals:
1. to test the competence of the candidate in a number of areas of either Computer Science or Mathematics
2. to evaluate the attitude and the potential of the candidate in carrying on research.

The admission exam **for graduates at foreign Universities** consists of an interview in which the candidate discusses a research project (written in English, max 500 words) presented with the application. The Commission may also ask questions on topics that are part of the courses of the bachelor degree ("laurea triennale" in Italian) in either Computer Science or Mathematics (see below for more details on the topics).

The admission exam **for graduates at Italian Universities** consists of a written exam and an oral exam. Both include questions on topics that are part of the courses of the bachelor degree ("laurea triennale" in Italian) in either Computer Science or Mathematics.
More specifically, at the written exam some topics are presented to the candidate (half topics in Computer Science and half topics in Mathematics) and the candidate has to select one of such topics and discuss it in the written text. The candidate interested in the Computer Science curriculum has to choose a Computer Science topic, while the candidate interested in the Mathematics curriculum has to choose a Mathematics topic.
At the oral exam, the Commission asks questions on other topics, but it may also require to further comment the written text. For the positions related to fellowships offered by companies on a specific project, the candidate will be also asked questions on topics of the specific project.

The topics of the questions (for graduates at Italian Universities as well as at foreign Universities) can be in the following areas of Mathematics:
- Algebra
- Geometry
- Mathematical Analysis
- Probability and Mathematical Statistics
- Mathematical Physics
- Numerical Analysis

as well as on the following areas of Computer Science:
- Theoretical computer science
- Computer architecture and operating systems
- Automata, formal languages and compilers
- Database systems
- Software engineering
- Artificial Intelligence
- Programming languages and paradigms
- Computer networks
- Human-Computer Interaction

The areas of Computer Science are better detailed in Appendix 1.

Appendix 1.

1) Theoretical computer science

- Abstract computational models (deterministic and non deterministic Turing machines, recursive functions)
- Computability (halting problem, simulation theorems)
- Computational complexity (cost models, asymptotic analysis and O-notation, complexity classes P, NP, EXP, logSPACE, PSPACE), completeness of a class, P=NP problem

Reference textbooks:

- J. Hopcroft, R. Motwani, J.D. Ullman. Introduction to automata theory, languages and computation. Third edition, Pearson, 2006 (also available in Italian).
- Ausiello, D'Amore & Gambosi. Linguaggi, modelli, complessità. Franco Angeli, 2003.
- T.H. Cormen, C.E. Leiserson, R.L. Rivest "Introduction to algorithms", Third edition, MIT Press, 2009. (also available in Italian).

2) Computer architecture and operating systems

Main concepts of computer architecture

- Computer Systems Organization: Multi-level machines.
- Digital Logic Level: basic combinatorial and sequential circuits
- The Microarchitecture Level.
- The Instruction Set Architecture Level (instruction format, addressing schemas, instruction types, control flow)
- Parallel Computer

Architectures Main concepts of operating systems:

- Process Management: Processes and Threads, CPU Scheduling, Process synchronization, Deadlocks
- Memory Management: Main Memory, Virtual Memory,
- Storage Management: File system, Mass-Storage structure, I/O systems

Reference textbooks:

- A.Tanenbaum: "Structured Computer Organization" 5th edition, Prentice Hall, 2006. (also available in Italian).
- Tanenbaum: "Modern Operating Systems, 2nd Ed.", Prentice Hall 2002 (also available in Italian).

3) Automata, formal languages and compilers

- Regular expressions, regular languages and finite automata.
- Context- free languages, push-down automata.
- Chomsky Hierarchy.
- Lexical analysis and syntax analysis.
- Syntax directed translation. Execution environments. Code generation.

Reference textbooks:

- J. Hopcroft, R. Motwani, J.D. Ullman. Introduction to automata theory, languages and computation. Second Edition, Pearson, 2001 (also available in Italian).

- A.V. Aho, R. Sethi e J.D. Ullman "Compilers Principles, Techniques, and Tools", Addison_wesley, 1985.

4) Database systems

- Data Models:High-level Conceptual Data Models for Database Design, Relational Data Model (Relational Algebra, Tuple Relational Calculus, Domain Relational Calculus and Visual Query Language, SQL).
- Database design
- Relational Normalization Theory.

Reference textbook:

- Paolo Atzeni, Stefano Ceri, Stefano Paraboschi and Riccardo Torlone. Database Systems - Concepts, Languages and Architectures. McGraw-Hill, 1999 (also available in Italian).

5) Software engineering

- Software life-cycle: Software process and software evolution.
- Software engineering principles and software quality
- Software design: Object-Oriented design.
- Unified Modeling Language.
- Software specification.
- Software verification.

Reference textbook:

- Carlo Ghezzi, Mehdi Jazayeri, Dino Mandrioli. Fundamentals of Software Engineering (2nd Edition), Prentice Hall, 2002 (also available in Italian).

6) Artificial intelligence

- Intelligent agents
- Automated problem solving (problem representation, solving problem by searching, heuristic search, problem decomposition, adversarial search)
- Knowledge representation: logical formalisms (propositional and first order-logic, automated inference)
- Knowledge representation: structured approaches (semantics nets, frames, ontologies)

Reference textbook:

- Russel Norvig: "Artificial Intelligence: A modern approach" 3rd edition, Prentice Hall, 2010 (also available in Italian)

7) Programming languages and paradigms

- Main concepts of programming languages: variables, expressions, data types, control structures, blocks and modules.
- Control abstractions: procedures.
- Data abstractions: abstract data types.
- Object oriented programming: objects, classes, inheritance, polymorphism.

Reference textbooks:

- Robert W. Sebesta. Concepts of Programming Languages. 9th edition, Addison Wesley, 2009
- M. Gabbrielli e S. Martini. “Linguaggi di programmazione: principi e paradigmi” McGraw- Hill Italia, 2006.

8) Computer networks

- Circuit and packet switching, delay, loss and throughput, protocol layers and the Internet protocol stack
- Application Layer protocols: HTTP, FTP, SMTP, DNS, socket programming.
- Transport Layer techniques and protocols: multiplexing-demultiplexing, connectionless transport and UDP, connection-oriented transport and TCP, TCP congestion control
- Networking layer and routing, IPv4, IPv6.
- Network security attacks, principles of cryptography, authentication, message integrity, access control, security across layers.

Reference textbook:

- J.F. Kurose, K.W. Ross. “Computer Networking: A Top-Down Approach, 5th Ed., Addison Wesley, 2009 (also available in Italian).

9) Human-Computer interaction

- Foundations
- Designing interaction
- Programming interactive systems
- User-centred design
- Usability and UX principles.
- Evaluation methods, user studies, usability metrics
- Prototyping

Reference textbooks:

- Preece, J., Rogers, Y., Sharp, H. "Interaction design, beyond human-computer interaction", John Wiley & Sons, 4th Edition, 2014.
- Polillo, R. "Facile da usare - Una moderna introduzione alla ingegneria dell'usabilità", Edizioni Apogeo, 2010.