Abilities summary

Florian Bonnard

Latest update : May 7, 2020

Contents

Chapter 1

Theoretical computer science and math

- Generic mathematics : probabilities (continuous, etc.), Markov chains and others ***(3rd year)***
- Advanced algebra : from in depth field theory to lattices with algorithms such as LLL algorithm ***(4th year)***
- Advanced algorithms : lattices, matroids, set cover, bin packing, k-center problem, metric uncapacitated facility localization problem, network flow problems (and generally speaking methods to model mathematical problems with algorithms) ***(3rd,4th year)***

- Logic and lambda calculus : general purpose logic through a variety of subjects with especially Curry-Howard isomorphism ***(4th year)***
- Semantics of programming languages : from basics with small steps semantics, big steps semantics and monads to advanced topics on modern typing systems (with both type verification and type inference) ***(4th,5th year + self-taught)***
- Formal verification / program proving : through different methods (abstracts domains, Hoare logic, weakest precondition) using frameworks such as Frama-C ***(5th year)***

- Complexity theory : generic complexity study (with complexity of quantum algorithms in 4th year) ***(3rd,4th year)***

- Quantum computer science : software design through quantum algorithms ***(4th year)***
- Cryptography : from the basics (RSA) to latest topics with elliptic curves ***(4th year)*** as well as several algorithms proposed for post-quantum cryptography standardization (NIST) ***(self-taught)***
- Error correcting codes : from simple linear codes to polar codes ***(3rd,4th year)*** with turbo codes and the study of some applications in 5G ***(self-taught)***
- Information theory : starting with Shannon theory with source-channel separation theorem ***(4th year)*** to more advanced topics such as rate distortion theory with Cover and Thomas book ***(self-taught)***

- Game theory : from 2 players to generalized theory ***(4th,5th year)***
- Optimization : generic optimization, combinatorial optimization, stochastic optimization, optimization applied to game theory ***(4th,5th year)***

- Distributed computing : generic distributed computing, self-stabilizing algorithms, distributed computing through combinatorial topology (especially on the study of possibility and impossibly of algorithms) ***(4th,5th year)***
- Graph theory : from generics (PageRank, etc.) to density-based clustering or hyperloglog algorithm ***(3rd,4th year)***
- Machine learning : from the generics (linear classifiers, etc.) to advanced topics on neural networks with a project on transfer learning ***(3rd,4th year)***

- Automata theory : from generics ones (finite-state machines) to more advanced ones (timed automatons, hybrid automatons, Büshi automatons) ***(3rd-5th year)***
- Abstract interpretation and abstract domains : from the generic model with simple examples (intervals) to more advanced examples with zonotopes and polytopes using ERAN and ELINA for implementations using C and Python mainly ***(5th year)***

- Network theory : from generics (such as OSI model and TLS protocol implementations) to advanced models based on stochastic processes ***(3rd,5th year)***
- Operating systems : a lot of topics including memory management (with alignment, etc.), file management and processes management ***(3rd year + self-taught)***

- Programming contest preparation for ICPC : generic preparation with focus on algorithms which are useful for the contest ***(4th year)***

Chapter 2

Applied computer science

- Generic development on Eclipse IDE ***(1st-5th year)***
- Generic development on OS-level virtualization technologies such as Docker to build containers ***(self-taught)***
- LaTeX report writing ***(self-taught)***
- Git for version control ***(3rd-5th year)***
- Generic cybersecurity with latest technologies : OWASP, web isolation, CASB, etc. with a formation by a Symantec engineer ***(5th year)***

2.1 Back-end

- Assembly languages : worked on different instruction sets but mainly on Intel processors ***(self-taught)***

- C language : general purpose programming ***(3rd year + self-taught)***
- C++ language : general purpose C++ before C++11 ***(4rd year)***, C++11 key concepts such as transfer semantics (with updated value categories from rvalue and lvalue to prvalue, xvalue and lvalue) and implementations of copy elision (with NRVO and RVO) ***(5th year)***, C++ 14 key concepts such as generic return type deduction ***(self-taught)***, C++17 many new concepts such as class template constructor deduction or guaranteed copy elision (though it should probably be called something like postponed temporary materialization) ***(self-taught)***, C++20 especially coroutines (which enables us to implement features similar to Python yield for instance) ***(self-taught)*** (yes I love C++)
- Parallel computing : through MPI and SIMD ***(5th year)*** and GPU programming with CUDA ***(self-taught)*** with C++ implementations (OpenMPI, MPICH and CUDA toolkit for C++)
- OpenGL ***(self-taught + implementation of a 3D model-projection-view algorithm for a project in 3rd year)***

- Linux / Windows : Unix Shell / DOS use ***(self-taught)***

- OCaml language and other ML languages (such as GhostML) : strongly-typed programming languages ***(4th,5th year)***

- PHP ***(self-taught)***
- Perl ***(self-taught)***

- Java : from the basics (Swing, etc.) to advanced topics such as garbage collector implementation or integration with C++ ***(3rd,4th year + self-taught)***
- Android ***(self-taught with project in 3rd year)***
- JDBC ***(5th year)***
- JSP (Java Server Pages) ***(5th year)***

- Web Services : SOAP and REST ***(5th year)***

- Python : with different libraries such as Sage (math), or Tensorflow and Scikit for machine learning ***(1st-5th year)***

- SQL / XML (with XPath, XQuery, XSLT, XSD, DTD) for database management ***(2nd,5th year + self-taught on the theoretical aspects)***

- Coq ***(self-taught)***
- Why3ML : language used under frameworks such as Frama-C ***(5th year)***
- Frama-C : language used for programs proofs in C ***(5th year)***

2.2 Front-end

- HTML : with HTML5 (written in XML style), and their definition from XML / SGML with advanced topics such as Shadow DOM and templates or generic selectors (following news from both W3C and WHATWG on hot topics especially on web components) ***(self-taught)***
- CSS with latest features such as Bootstrap as well as their implementation details (with SCSS and SASS pre-processors like the one of Sublime Text) ***(self-taught)***

- JPA (Java Persistence API) ***(self-taught)***
- Javascript and Javascript frameworks like Node.js (with most of its basics libraries mastered (streams, sockets, http (http/1.1 and http/2 and waiting for http/3 full implementation))), jQuery, React.js and Angular.js ***(self-taught)***
- Following V8 (Google) vs SpiderMonkey (Firefox) development ***(self-taught)***
- Ruby / Ruby on Rails : The entire Ruby language (up to 2.7.0) as well as the Ruby on Rails framework (in the context of use with Docker in particular) ***(self-taught)***

Chapter 3

Physics

- Physics generics : really good understanding of theoretical physics in general (major in 2nd year) and still used nowadays to understand implementation of norms such as USB, Wi-Fi, etc. in computer science ***(1st,2nd year)***

- Optics and photonics ***(3rd year)***
- Wave propagation ***(3rd year)***
- Digital communications ***(3rd year)***
- Electronic acquisition systems ***(3rd year)***
- Processors and digital architectures ***(3rd year)***
- Micro and nano physics ***(3rd year)***

Chapter 4

Engineering sciences

- All the basics with a really good theoretical understanding (2^{nd} major in 2^{nd} year) ***(1^{st},2^{nd} year)***
- Still following a lot of actualities on the topics linked to engineering sciences (through implementations of computing technologies) ***(self-taught)***

Chapter 5

Miscellaneous

- Agile software development (Scrum) ***(4th,5th year)***

- Introduction to management : study of how to take decisions in a production company (phone producer in our concrete example) in a team through a concrete example ***(3rd year)***
- Introduction to modern economy : introduction to modern economy concepts (especially demand, supply and balance through concepts such as elasticity) and study of a chosen economy article (I studied prospect theory and how it challenges the usual expected utility theory through risk aversion (combination of economics, math and psychology)) ***(3rd year)***
- Corporate finance : study of fundamentals of finance in companies (especially how to read and use the three important financial statements : income statement, balance sheet and cash flows) and how they are applied in the context of stock exchange with a concrete case study ***(4th year)***

- Laws (especially Europeans) linked to technologies : General Data Protection Regulation (GDPR) and its applications in companies / organizations ***(4th year)***
- Information and communication technologies challenges : carrying out a collective inquiry on a scientific or technical controversy and publishing the results in the form of a website (the concrete case was glyphosate, and I worked mainly on the law part at an international level) ***(3rd year)***
- Activism in the digital era : study of how oneself can act for his social or environmental opinions in the context of politics (particularly linked to the application of GDPR by non-profitable organizations) ***(4th year)***
- Digital sociology : generic introduction + project on the role of algorithms in our daily life ***(4th year)***

- Written communication practices and analysis : study of how to produce a concrete production on a topic which is important to us during 6 months (I choose to work on the role of technologies in education and presented it through a website) ***(3rd year)***

- Humor in work relationship (funnier than I initially though) : study of subjects which deal with communication with teams at work ***(4th year)***

- Philosophy of sciences : study of the methods of reasoning through the centuries (from the 10th century to nowadays) ***(4th year)***