Course catalog proposal – Spring semester (February – June)

Course Name	Level	Lectures (h)	Tutorials (h)	Practical work (h)	ECTS
Research project (advanced)	Master/Graduate		20		5
Project in Flows, Production, Graphs, Operational Research	Master/Graduate		4	32	3
Flows and Production Management	Master/Graduate	12	10	4	3
AI, graphs and applications	Master/Graduate	16	16		3
Operational Research and Applications	Master/Graduate	12	8	12	3
Object Oriented algorithms and programming (C++)	Bachelor/Undergrad.	16	8	8	4
C++ Project	Bachelor/Undergrad.			32	3
Distributed Systems and Programming	Bachelor/Undergrad.	18	8	20	4
Object Oriented modeling and programming (Java)	Bachelor/Undergrad.	14	8	10	4
Integrated Development Environments (C++)	Master/Graduate	4		20	3
IS option: introduction to Information Systems and to their architecture	Master/Graduate	10	10	24	3
SA option: Distributed computing	Master/Graduate	8	6	6	2
Team project	Master/Graduate			10	6

Course details – Spring semester (February - June)

Research project (advanced) Course code: S10.1 Lectures: 0h Tutorials: 20h Practical work: 0h ECTS: 5 Course supervisor: Project supervisor	This project consists in studying a research problem and in implementing/testing a possibly new solution to such problem
Project in Flows, Production, Graphs, Operational Research Course code: S10.O10 Lectures: 0h Tutorials: 4h Practical work: 32h ECTS: 3 Course supervisor: Project supervisor	The student chooses a topic between those covered by units S10.04, S10.05, or S10.07, namely, production management, graphs, or operations research. Then, under the advice of a faculty, the student will study a problem to which s/he should propose and implement a solution.
Flows and Production Management Course code: S10.O4 Lectures: 12h Tutorials: 10h Practical work: 4h ECTS: 3 Course supervisor: Vincent T'Kindt	This course introduces the basics of production management: general introduction, inventory management (deterministic models EOQ, EPQ, stochastic models), MRP and MRP II and algorithms for solving scheduling problems.

AI, graphs and applications Course code: S10.O5 Lectures:16h Tutorials: 16h Practical work: 0h ECTS: 3 Course supervisor: Nicolas Ragot	The graph matching part of this course gives students an overview of graph based representations of complex objects like image contents and methods for graph comparison. It includes some classical algorithms of exact graph matching like sub-graph isomorphism and some more sophisticated algorithms (error tolerant graph matching) like Graph Edit Distance computation or Graph Embedding techniques. Those algorithms (and others) are illustrated and applied for image segmentation, data classification, pattern recognition or interactive data visualization tasks.
Operational Research and Applications Course code: S10.O7 Lectures: 12h Tutorials: 8h Practical work: 12h ECTS: 3 Course supervisor: Ameur Soukhal	This lecture provides an overview of operations research (OR) from the perspective of an industrial engineer. The focus of this lecture is on the "OR approach" to solving design and operational problems that industrial engineers commonly encounter. In its most basic form, OR may be viewed as a scientific approach to solving problems; it abstracts the essential elements of the problem into a model, which is then analyzed to yield an optimal solution for implementation. The mathematical details and the specific techniques used to build and analyze these models can be quite sophisticated and are addressed; the emphasis of this lecture is on the approach. Detailed description of methodology of OR based on graph theory and linear programming methods will be presented. This lecture concludes with some examples of successful real world applications of OR.
Object Oriented algorithms and programming (C++) Course code: S6.2.1&2 Lectures: 16h Tutorials: 8h Practical work: 8h ECTS: 4 Course supervisor: Vincent T'Kindt	This course is devoted to software development under the object oriented paradigm. Three topics are covered: algorithms, modeling, and object-oriented languages. The first part presents the basic notions of objects (heritage, overload, polymorphism...). The second part introduces the unified modeling language (UML) methodology. Finally, the third part discusses two object oriented programming languages: C++ and Java.

C++ Project Course code: S6.2.3 Lectures: 0h Tutorials: 0h Practical work: 32h ECTS: 3 Course supervisor: Project supervisor	In this project, the student should provide an object oriented programming based solution to a predefined problem (in C++). As part of the project, the student should develop a prototype and test his or her implementation.
Distributed Systems and Programming Course code: S6.4 Lectures:18h Tutorials: 8h Practical work: 20h ECTS: 4 Course supervisor: Mathieu Delalandre	The first part of this course provides an introduction to distributed systems (relations to parallel systems, characterization, new trends, goals and challenge, etc.) and then discusses the inter-process communication (IPC) model (socket, stream and message oriented communication, group communication, MPI, message queuing and IDL). The course sets a particular focus on network programming; the UDP and TCP Java interfaces are investigated during the practical sessions. The second part of this course introduces the student to parallel programming. The course starts by introducing parallel computing, its applications, its benefits, and its limitations. Then, the course introduces the principles of parallel algorithm design and discusses analytical modeling of parallel programs. Next, the course discusses synchronizers (e.g., locks and semaphores) and their applications. Finally, the course introduces the Java concurrent package and studies its main components. The course sets a special focus on practical work on the design and implementation of parallel algorithms in Java.
Object Oriented modeling and programming (Java) Course code: S6.5.1&2 Lectures:14h Tutorials: 8h Practical work: 10h ECTS: 4 Course supervisor: Vincent T'Kindt	This course is devoted to software development under the object oriented paradigm. Three topics are covered: algorithms, modeling, and object-oriented languages. The first part presents the basic notions of objects (heritage, overload, polymorphism...). The second part introduces the unified modeling language (UML) methodology. Finally, the third part discusses two object oriented programming languages: C++ and Java.

Integrated Development Environments (C++) Course code: S8.2.1 Lectures:4h Tutorials: 0h Practical work: 20h ECTS: 3 Course supervisor: Yannick Kergosien	To follow this course, basic notions of C++ language are required. The course is divided into two parts. The first part deals with the STL library that provides a collection of classes and functions for generic programming (e.g. containers, function objects, generic strings and streams). The second, and main, part is devoted to the famous cross platform application framework: Qt. After introducing the principal components of Qt (i.e. QObject, signals and slots, Metaobject compiler), the course discusses Qt Widgets and the Qt Creator platform. The notions of the Model/View/Controller pattern are cover through the development of an application. The evaluation is based on a project that should be developed all along the practical work sessions.
IS option: introduction to Information Systems and to their architecture Course code: S8.3a.1&2 Lectures:10h Tutorials: 10h Practical work: 24h ECTS: 3 Course supervisor: Donatello Conté	The objectives of this courses are the following: 1) Master the classical architectures of IS, in their conception, their implementation, their use 2) Have notions of IS security 2) Know the principles of IS urbanization 3)Know how to describe all the business processes and activities of the company that the IS must support(business vision), 3) To be able to offer a structuring framework for the processes 4)the functions in communicating functional blocks (functional vision), 5) to be able to define the applications that automate the functions, and the technical infrastructure enabling their exploitation (computer vision).
SA option: Distributed computing Course code: S8.4b.2 Lectures:8h Tutorials: 6h Practical work: 6h ECTS: 2 Course supervisor: Mathieu Delalandre	This lecture is related to general aspects of distributed computing. The course addresses different issues including introduction about the topic, physical clock synchronization, event synchronization and global states, coordination and agreement including the distributed mutual exclusion, multicast communication and election problems. These aspects are investigated at the concept and principle levels during tutorials, and through applications during practical work sessions.

Team project Course code: S8.5 Lectures: 0h Tutorials: 0h Practical work: 64h ECTS: 6 Course supervisor: Project supervisor	This project is performed in groups of about 6 students. It consists in using all the acquired knowledge to develop software for solving a given problem. The group must also use all the software engineering concepts.