Subjects Modules for S1

Semester 1 Year 2

U3.1: Optimization

Language Theory and Compilation

Module designation	**Optimization**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.1
Subtitle, if applicable	
Courses, if applicable	**Language Theory and Compilation**
Semester(s) in which the module is taught	Semester3
Person responsible for the module	Dept Head
Lecturer	Ms. Lamia Ghezail
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours of Classroom Lecture/ Semester
Workload	Total 42 hours/ Semester (21 hours of Self Study)
Credit points	2
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Advanced coursework in programming languages and algorithms. Thorough understanding of formal language theory. Proficiency in a high-level programming language (e.g., Java, C++)
Module objectives/intended learning outcomes	**Learning Outcomes:** Upon completion of the course, students should be able to: • Language Understanding: Develop a deep understanding of formal languages, grammars, and language hierarchies, including context-sensitive languages. • Compiler Design: Design and implement compilers for complex programming languages, encompassing lexical analysis, parsing, semantic analysis, optimization, and code generation. • Optimization Techniques: Apply advanced optimization techniques to enhance the efficiency and performance of generated code. • Error Handling: Implement advanced error detection, recovery, and correction mechanisms, ensuring robustness in the face of errors. • Language Formalization: Formally define and analyze the syntax and semantics of programming languages using cutting-edge formal methods and semantic frameworks.

Content	**Chapter 1: Introduction to Language Theory and Compilation** 1. Overview of the compilation process. 2. Formal language theory fundamentals. 3. Historical context of compiler development. 4. Emerging trends in language design. **Chapter 2: Lexical Analysis** 1. Regular expressions and finite automata. 2. Lexical analysers and tokenization. 3. Challenges in handling complex lexemes. 4. Practical examples and case studies. **Chapter 3: Syntax Analysis** 1. Context-free grammars and parsing techniques. 2. Syntax-directed translation and abstract syntax trees. 3. Parsing table generation and optimization. 4. Advanced parsing algorithms. **Chapter 4: Semantic Analysis** 1. Semantic analysis and type checking. 2. Symbol tables, scope analysis, and resolution. 3. Handling polymorphism and type inference. 4. Advanced techniques in semantic analysis. **Chapter 5: Intermediate Code Generation** 1. Three-address code generation. 2. Intermediate code optimization techniques. 3. Representing control flow in intermediate code. 4. Case studies of real-world intermediate code. **Chapter 6: Code Generation and Optimization** 1. Code generation for target machines. 2. Code optimization strategies for maximizing code efficiency. 3. Register allocation and addressing modes. 4. Profile-guided optimization and dynamic compilation. **Chapter 7: Error Handling and Recovery** 1. Error detection, reporting, and recovery strategies. 2. Advanced techniques for error correction and resilience. 3. Design principles for error-resistant compilers. 4. Case studies in error handling.

	Chapter 8: Programming Language Formalization 1. Formal specification of programming languages. 2. Denotational and operational semantics. 3. Advanced semantic frameworks and theorem proving. 4. Practical applications of language formalization
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & online (Moodle ULT) Video projection
Reading list	1. Muchnick, S. S. (1997). "Advanced Compiler Design and Implementation." Morgan Kaufmann. 2. Appel, A. W. (2002). "Modern Compiler Implementation in C/Java/ML." Cambridge University Press. 3. Cooper, K. D., & Torczon, L. (2011). "Engineering a Compiler." Morgan Kaufmann. 4. Nielson, F., & Nielson, H. R. (2012). "Semantics with Applications: An Appetizer." Springer.

U3.1: Optimization

Graph theory

Module designation	**Optimization**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.1
Subtitle, if applicable	
Courses, if applicable	**Graph theory**
Semester(s) in which the module is taught	Semester 3
Person responsible for the module	Dept Head
Lecturer	Ms. Amira Brahmi
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours of Classroom Lecture/ Semester
Workload	Total 42 hours/ Semester (21 hours of Self Study)
Credit points	2
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Advanced coursework in programming languages and algorithms. Thorough understanding of formal language theory. Proficiency in a high-level programming language (e.g., Java, C++)
Module objectives/intended learning outcomes	**Objectives:** • To introduce students to the fundamental principles of graph theory. • To develop the ability to model systems and problems using graphs. • To provide algorithmic approaches to solving graph-based problems. • To enable the application of graph theory in computer science, data science, and engineering domains. **Learning Outcomes:** By the end of this course, students will be able to: • Understand fundamental concepts and definitions in graph theory. • Analyze the structural properties of different types of graphs. • Apply graph traversal algorithms in various contexts.

	• Solve problems related to connectivity, cycles, and trees. • Use graph coloring techniques in real-world applications. • Determine planarity and identify planar embeddings. • Design and analyze algorithms for shortest paths, matchings, and spanning trees. • Model and solve real-life problems using graph-theoretical concepts.
Content	**1. Introduction to Graphs** 1.1 Basic definitions and concepts 1.2 Types of graphs (simple, directed, weighted, multigraphs, etc.) **2. Graph Representations** 2.1 Adjacency matrix 2.2 Incidence matrix 2.3 Adjacency list **3. Graph Isomorphism and Subgraphs** 3.1 Isomorphic graphs 3.2 Subgraphs and induced subgraphs 3.3 Graph invariants **4. Connectivity** 4.1 Connected graphs and components 4.2 Cut vertices, bridges 4.3 Menger's Theorem **5. Eulerian and Hamiltonian Graphs** 5.1 Eulerian paths and circuits 5.2 Hamiltonian paths and cycles 5.3 Applications **6. Trees and Forests** 6.1 Definition and properties 6.2 Spanning trees 6.3 Minimum spanning trees (Prim's and Kruskal's algorithms) **7. Directed Graphs and DAGs** 7.1 Directed graphs 7.2 Topological sorting 7.3 Strongly connected components **8. Graph Algorithms** 8.1 BFS, DFS 8.2 Dijkstra's algorithm

	8.3 Bellman-Ford and Floyd-Warshall algorithms
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & online (Moodle ULT) Video projection
Reading list	1. Bondy, J.A., & Murty, U.S.R. (2008). Graph Theory. Springer. 2. Diestel, R. (2017). Graph Theory (5th ed.). Springer. 3. West, D.B. (2001). Introduction to Graph Theory (2nd ed.). Prentice Hall. 4. Gross, J.L., & Yellen, J. (2005). Graph Theory and Its Applications (2nd ed.). CRC Press. 5. Cormen, T.H., Leiserson, C.E., Rivest, R.L., & Stein, C. (2009). Introduction to Algorithms (3rd ed.). MIT Press. (For graph algorithms)

U3.1: Optimization

Complexity and Optimization of Algorithms

Module designation	**Optimization**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.1
Subtitle, if applicable	
Courses, if applicable	**Complexity and Optimization of Algorithms**
Semester(s) in which the module is taught	Semester 3
Person responsible for the module	Dept Head
Lecturer	Ms. Kadria Ezzine
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	Lecture, 31.5 hours of Classroom Lecture/ Semester
Workload	Total 66.5 hours/ Semester (35 hours of Self Study)
Credit points	3
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Algorithmics and Data Structures Basic Math
Module objectives/intended learning outcomes	**Objectives:** This course introduces basic elements of the design and analysis of computer algorithms. Topics include asymptotic notations and analysis, divide and conquer strategy, greedy methods, dynamic programming, basic graph algorithms, NP-completeness, and approximation algorithms. For each topic, beside in-depth coverage, one or more representative problems and their algorithms shall be discussed. **Learning Outcomes:** Students will be able to: 1. Analyze the asymptotic performance of algorithms. 2. Demonstrate a familiarity with major algorithm design techniques 3. Apply important algorithmic design paradigms and methods of analysis. 4. Solve simple to moderately difficult algorithmic problems arising in applications. 5. Able to demonstrate the hardness of simple NP-complete problems

Content	**Chapter 1: Foundations of Algorithm Analysis** 1. Algorithms and its properties 2. Time and Space Complexity 3. Detailed Analysis of algorithms 4. Concept of Aggregate Analysis 5. Asymptotic Notations 6. Recursive Algorithms and Recurrence Relations **Chapter 2: Iterative Algorithms** 1. Basic Algorithms Algorithm for GCD, Fibonacci Numbers and analysis of their time and space complexity 2. Searching Algorithms: Sequential Search and its analysis 3. Sorting Algorithms: complexity analysis. **Chapter 3: Case Study** 1. Divide and Conquer Algorithms 2. Greedy Algorithms 3. Dynamic Programming 4. Backtracking **Chapter 4: Number Theoretic Algorithm** 1. Concept of Number Theoretic Notation 2. Concept of Modular Linear Equations 3. Euclid's and Extended Euclid's Algorithms for solving Modular Linear Equations. **Chapter 5: NP Completeness** 1. Tractable and Intractable Problems, Complexity Classes 2. NP Complete Problems 3. Approximation Algorithms
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online(Moodle ULT) Video projection
Reading list	1. Herbert S. Wilf, Algorithms and Complexity, A K Peters/CRC Press; 2nd edition, 2002 Preliminaries on algorithms and Complexity. 2. Schrijver, Alexander, Combinatorial Optimization: Polyhedra and Efficiency. Vol. 1. Springer, 2003. 3. Levitin, A. Introduction to the design & analysis of algorithms. Boston: Pearson, 2012. 4. Cormen, T. H.,Introduction to algorithms. Cambridge, MA: MIT Press. 2009.

	5. Kleinberg, J., & Tardos, E. Algorithm design. Boston: Pearson/Addison-Wesley, 2006.

U3.2: Web Programming

Advanced web programming

Module designation	**Web Programming**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.2
Subtitle, if applicable	
Courses, if applicable	**Advanced web programming**
Semester(s) in which the module is taught	Semester 3
Person responsible for the module	Dept Head
Lecturer	Mr. Nassim Bahri
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester 21 hours of Supervised projects on Campus/ semester
Workload	Total 77 hours/ Semester (35 hours of Self Study)
Credit points	3
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Basic knowledge of PHP programming. Understanding of Object-Oriented Programming (OOP) concepts. Familiarity with web development basics (HTML, CSS, JavaScript).
Module objectives/intended learning outcomes	**Objectives:** The objective of this course is to provide students with practical skills in building modern web applications using the Symfony framework. It focuses on mastering the MVC architecture, routing, controllers, and templating with Twig. Students will learn to manage databases with Doctrine, implement security features, and build RESTful APIs. The course also emphasizes best practices, code organization, and deployment techniques. **Learning Outcomes:** Students will be able to: 1. Understand the structure and components of the Symfony framework. 2. Develop and deploy web applications using Symfony. 3. Integrate databases and manage data using Doctrine. 4. Implement authentication and authorization techniques in Symfony. 5. Deploy Symfony application.

Content	**Chapter 1: Introduction to Symfony** 1. Basics of Symfony 2. Symfony versus other PHP frameworks 3. Modern PHP (namespaces, composer, etc) 4. Installing Symfony **Chapter 2: Symfony Structure and Components** 1. Symfony directory structure 2. Key Symfony Components (HttpFoundation, Routing, etc.) **Chapter 3: Developing Web Applications** 1. Controllers, Routes, and Views 2. Twig templating engine **Chapter 4: Database Integration with Doctrine** 1. Configuring databases 2. CRUD operations 3. Data relationships **Chapter 5: Symfony Forms & Validation** 1. Symfony Forms 2. Symfony Validation **Chapter 6: Security in Symfony** 1. User authentication 2. Authorization and roles **Chapter 7: Project deployment**
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Project (50%)(Report for each workshop/Project required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online(Moodle ULT) Video projection
Reading list	1. Fabien Potencier (2017). Symfony 5: The Fast Track. Symfony SAS. 2. Bernhard Schussek, Ryan Weaver (2015). Symfony Certification: Study Guide. SensioLabs. 3. KnpUniversity.com. Symfony screencasts and tutorials. 4. Official Symfony Documentation. Available at https://symfony.com/doc/current/index.html.

U3.2: Web Programming

JEE Technologies

Module designation	**Web Programming**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.2
Subtitle, if applicable	
Courses, if applicable	**JEE Technologies**
Semester(s) in which the module is taught	**Semester 3**
Person responsible for the module	Dept Head
Lecturer	Mr. Nassim Bahri
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester 21 hours of Supervised projects on Campus/ semester
Workload	Total 77 hours/ Semester (35 hours of Self Study)
Credit points	3
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Understanding of Object-Oriented Programming (OOP) concepts. Familiarity with web development basics (HTML, CSS, JavaScript).
Module objectives/intended learning outcomes	**Course Objectives:** This course will enable students to: • Construct client-server applications using Java socket API • Adapt servlets to build server-side programs • Demonstrate the use of JavaBeans to develop component-based Java software **Learning outcomes:** The student will be able to: • Interpret the need for advanced Java concepts like enumerations and collections in developing modular and efficient programs • Build client-server applications and TCP/IP socket programs • Describe how servlets fit into Java-based web application architecture

	• Develop reusable software components using Java Beans
Content	**1.** JEE Architecture **2.** Remote Method Invocation **3.** Implement server-side programming using Servlets and Implement various session management techniques **4.** Handle errors and exceptions in Servlets Application **5.** Implement inter-Servlets Communication **6.** Develop JSP Applications **7.** Web Application Basics **8.** Struts Architecture **9.** Action Mappings **10.** Forms **11.** Struts Tag Libraries **12.** The JSP Standard Tag Library **13.** Spring Framework **14.** Hibernate **15.** Object/Relational Mapping **16.** The Criteria Query API **17.** Hibernate Query Language **18.** Session Handling in Web Application ***Workshop in Lab*** ***Group Projects are proposed***
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Project (50%) (Report for each workshop/Project required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online(Moodle ULT) Video projection
Reading list	1. Y. Daniel Liang: Introduction to JAVA Programming, 7thEdition, Pearson Education, 20 2. Herbert Schildt: JAVA the Complete Reference, 7th/9th Edition, Tata McGraw Hill, 2007. 3. Jim Keogh: J2EE-TheCompleteReference, McGraw Hill, 2007.

U3.3: Advanced Databases

Distributed Databases

Module designation	**Advanced Databases**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.3
Subtitle, if applicable	
Courses, if applicable	**Distributed Databases**
Semester(s) in which the module is taught	**Semester 3**
Person responsible for the module	Dept Head
Lecturer	Mr. Imed Hammadi
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours Lecture/ semester 21 hours practical workshop in Lab/ semester
Workload	Total 77 hours/ Semester (35 hours of Self Study)
Credit points	3
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Database fundamentals
Module objectives/intended learning outcomes	**Course Objectives:** The objective of this course is to introduce students to the principles, architecture, and challenges of distributed database systems. It focuses on data distribution, query optimization, transaction management, replication, and consistency. Students will gain theoretical understanding and practical skills in designing and managing distributed databases, including modern NoSQL systems. **Learning Outcomes:** By the end of this course, students will be able to: • Understand the key concepts and architecture of distributed databases • Design fragmentation and data allocation strategies • Manage distributed transactions using appropriate concurrency and commit protocols

	• Apply distributed query optimization techniques • Evaluate consistency models and replication strategies
Content	**Chapter 1: Introduction to Distributed Databases** • Definition, motivations, and key challenges • Differences between centralized and distributed databases **Chapter 2: Distributed Database Architecture** • Client-server, peer-to-peer, and multi-tier architectures • Transparency features: location, replication, fragmentation **Chapter 3: Data Fragmentation and Allocation** • Horizontal and vertical fragmentation • Data allocation strategies and optimization **Chapter 4: Distributed Query Processing** • Query decomposition and optimization • Cost models and efficient execution plans **Chapter 5: Distributed Transaction Management** • ACID properties in distributed systems • Two-phase commit protocol (2PC) • Concurrency control techniques **Chapter 6: Distributed Concurrency Control** • Locking, timestamp ordering, and multi-version techniques • Deadlock detection and resolution **Chapter 7: Replication and Consistency** • Replication models and strategies • Consistency models: strong, eventual, causal **Chapter 8: Distributed Database Security** • Authentication, authorization, and secure communication
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online(Moodle ULT) Video projection

Reading list	1. Özsu, M. Tamer & Valduriez, Patrick – Principles of Distributed Database Systems, Springer 2. Elmasri, Ramez & Navathe, Shamkant B. – Fundamentals of Database Systems, Pearson (selected chapters on distributed databases) 3. Coulouris, G., Dollimore, J., Kindberg, T., & Blair, G. – Distributed Systems: Concepts and Design, Pearson

U3.3: Advanced Databases

Data Mining

Module designation	**Advanced Databases**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.3
Subtitle, if applicable	
Courses, if applicable	**Data Mining**
Semester(s) in which the module is taught	**Semester 3**
Person responsible for the module	Dept Head
Lecturer	Mr. Mehdi Kaabi
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours of Classroom Lecture/ Semester 21 hours practical workshop in Lab/ semester
Workload	Total 63 hours/ Semester (21 hours of Self Study)
Credit points	3
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Object Oriented Programming Artificial Intelligence Basics
Module objectives/intended learning outcomes	**Course Objectives:** The objective of this course is to provide students with a comprehensive understanding of data mining techniques and their applications. The course focuses on the process of discovering patterns, knowledge, and insights from large datasets using statistical, machine learning, and algorithmic approaches. Students will learn to preprocess data, apply core data mining methods, and evaluate results in real-world contexts. **Learning outcomes:** The student will be able to: • Understand the data mining process and its role in knowledge discovery • Perform data preprocessing including cleaning, transformation, and reduction • Apply techniques such as classification, clustering,

	association rule mining, and anomaly detection • Evaluate data mining models using appropriate metrics and validation methods • Use data mining tools and software (e.g., WEKA) • Understand ethical considerations and challenges in data mining
Content	**Chapter 1: Advanced Understanding of Data Mining** 1. Classification 2. Regression 3. Clustering 4. association methods **Chapter 2: Mastery of Data Mining Tools** 1. advanced data mining tools such as R, Python (scikit-learn, pandas), WEKA 2. commercial solutions for data analysis **Chapter 3: Advanced Data Preparation** 1. collection 2. Cleaning 3. Transformation 4. integration **Chapter 4: Advanced Data Mining Algorithms** 1. support vector machines (SVM) 2. neural networks 3. advanced decision trees 4. ensemble methods **Chapter 5: Evaluation of Data Mining Models** 1. advanced metrics 2. area under the ROC curve (AUC)

	3. recall 4. F1-score
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online(Moodle ULT) Video projection
Reading list	1. Han, J., Kamber, M., & Pei, J. – Data Mining: Concepts and Techniques (4th ed.), Morgan Kaufmann 2. Tan, P.-N., Steinbach, M., Karpatne, A., & Kumar, V. – Introduction to Data Mining (2nd ed.), Pearson 3. Witten, I.H., Frank, E., Hall, M.A., & Pal, C.J. – Data Mining: Practical Machine Learning Tools and Techniques, Morgan Kaufmann 4. Aggarwal, C.C. – Data Mining: The Textbook, Springer

U3.4: Big Data & Cloud

Big Data

Module designation	**Big Data & Cloud**
Module level, if applicable	Year 3, Semester 1
Code, if applicable	U3.4
Subtitle, if applicable	
Courses, if applicable	**Big Data**
Semester(s) in which the module is taught	Semester 3
Person responsible for the module	Dept Head
Lecturer	Ms. Salma Bouazizi
Language	English
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours practical workshop in Lab/ semester 21 hours for Projects/ Semester
Workload	Total 84 hours/ Semester (42 hours of Self Study)
Credit points	3
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Programming Language (Java), Practice of SQL (queries and sub queries), exposure to Linux Environment
Module objectives/intended learning outcomes	**Objectives:** Acquire fundamental practical knowledge of the different concepts, techniques and products related to Big Data Learn how to store, manage, process and analyze large sets of unstructured data **Learning outcomes:** At the end of the course students will be able to: 1. Identify Big Data and its Business Implications. 2. List the components of Hadoop and Hadoop Eco-System 3. Access and Process Data on Distributed File System 4. Manage Job Execution in Hadoop Environment 5. Develop Big Data Solutions using Hadoop Eco System 6. Analyze Infosphere BigInsights Big Data Recommendations. Apply Machine Learning Techniques using R.

Content	**I – Introduction** 1- The four dimensions of Big Data: volume, velocity, variety, veracity 2- Presentation of the MapReduce set, storage and queries **II- Improve business results with Big Data** 1- Measure the importance of Big Data within a company 2- Succeed in extracting useful data 3- Integrate Big Data with traditional data **III- Analyse the characteristics of your data** 1- Select the data sources to analyse 2- Remove duplicates 3- Define the role of NoSQL **IV- Presentation of Big Data warehouses** 1- Data models: key value, chart, document, column family 2- Hadoop Distributed File System (HDFS) **V- Choose a Big Data warehouse** **VI- Integrate different data warehouses** 1- Map the data with the programming framework, connect to the data and 2- Extract from the storage warehouse, transform the data to be processed 3- Split data for Hadoop MapReduce **VII- Using Hadoop** **VIII- Extract data giving value to the company** **IX- Develop a strategy dedicated to Big Data** 1- Define Big Data needs 2- Achieving the objectives thanks to the relevance of the data 3- Evaluate the different tools on the market dedicated to Big Data 4- Meet the expectations of the company's staff **X- Statistical analysis of Big Data** 1- Exploit the RHadoop functionality 2- Generate statistical reports with Rhadoop 4- Use RHadoop visualization 5- Use the results of the analyses ***Projects: Implement a Big Data solution***
Study and examination requirements and forms of examination	Continuous Assessment (50%) + Evaluation of the final report of project (50%) (Report for each workshop/Project required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online(Moodle ULT) Video projection
Reading list	1. Michael Berthold, David J. Hand, "Intelligent Data Analysis", Springer, 2007. 2. Jay Liebowitz, "Big Data and Business Analytics" Auerbach Publications, CRC press (2013) Tom Plunkett, Mark Hornick, "Using R to Unlock the Value of Big Data: Big Data Analytics with Oracle R Enterprise and Oracle R Connector for Hadoop", McGraw-Hill/Osborne Media (2013), Oracle press.

U3.4 Big Data & Cloud

Virtualization and Cloud Computing

Module designation	**Big Data & Cloud**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.4
Subtitle, if applicable	
Courses, if applicable	**Virtualization and Cloud Computing**
Semester(s) in which the module is taught	**Semester 3**
Person responsible for the module	Dept Head
Lecturer	Mr. Mahdmod Samrani
Language	English
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours Lecture/ semester 21 hours practical workshop in Lab/ semester
Workload	Total 77 hours/ Semester (35 hours of Self Study)
Credit points	3
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	Operating System (Linux), Computer Network
Module objectives/intended learning outcomes	**Course Objectives:** The course presents a top-down view of cloud computing, from applications and administration to programming and infrastructure. Its main focus is on parallel programming techniques for cloud computing and large-scale distributed systems which form the cloud infrastructure. The topics include overview of cloud computing, cloud systems, parallel processing in the cloud, distributed storage systems, virtualization, security in the cloud, and multicore operating systems. Students will study state-of-the-art solutions for cloud computing developed by Google, Amazon, Microsoft, Yahoo, VMWare, etc. Students will also apply what they learn in one programming assignment and one project executed over Amazon Web Services. **Learning outcomes:** 1. Analyze the trade-offs between deploying applications in the cloud and over the local infrastructure. 2. Compare the advantages and disadvantages of various cloud computing platforms. 3. Deploy applications over commercial cloud computing infrastructures such as Amazon Web Services, Windows Azure, and Google AppEngine.

	4. Program data intensive parallel applications in the cloud. 5. Analyze the performance, scalability, and availability of the underlying cloud technologies and software. 6. Identify security and privacy issues in cloud computing. 7. Explain recent research results in cloud computing and identify their pros and cons. 8. Solve a real-world problem using cloud computing through group collaboration.
Content	**Chapitre 1: Introduction to computer systems** 1. Operating system 2. Computer servers 3. Computer Security 4. Distributed systems **Chapitre 2: Virtualisation** 1. Definition of Virtualization 2. Domains of virtualization 3. Advantages & disadvantages of virtualization 4. Different types of virtualizations **Practical Work:** - TP1: Master the different types of virtualization hypervisor, the principle of type 2 virtualization. - TP2: Master the basic concepts of type 1 virtualization. **Chapitre 3: Cloud Computing** 1. IT Challenges 2. Definition and characteristics of the Cloud 3. Cloud architecture, models and services 4. Security and privacy 5. Economic issues 6. Research axes **Practical Work:** - TP3: Become familiar with Cloud Computing IaaS environments (Virtualbox & OpenStack integration) - TP4: Study and manipulate the advanced features of the OpenStack environment. **Chapitre 4: Cloud Services Security** 1. Cloud Security 2. National Institute of Standard for Technology (NIST) recommendations 3. Cloud Security Alliance (CSA) recommendations **Chapitre 5: Docker** 1. Concepts 2. Setup 3. Create and use containers 4. Use and share images 5. Network management 6. Volume management 7. Docker Composer **Practical Work:** - TP5: Understand and test the main uses of dockers,

	master the creation of images and dockerfiles. - TP6: Master the management of networks and volumes of dockers
Study and examination requirements and forms of examination	Written Mid-term Exam (25%) + Practical Exam (25%) + Written Final Exam (50%) (Report for each workshop required)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT) Video projection
Reading list	1. Cloud Computing: From Beginning to End, By Ray J. Rafaels, April 2015 2. Cloud Computing: Concepts, Technology & Architecture and Cloud Computing Design Patterns, Thomas Erl and others, May 2013 3. Cloud Computing, Sécurité, stratégie d'entreprise et panorama du marché, Guillaume Plouin, Edition DUNOD, 2013 4. Cloud Computing, Maîtrisez la plateforme AWS, Amazon Web Services, Mathieu Zarrouk, Edition ENI, 2012

U3.5 Soft skills & Languages I

Scientific Writing

Module designation	**Soft skills & Languages I**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.5
Subtitle, if applicable	-
Courses, if applicable	Scientific Writing
Semester (s) in which the module is taught	Semester 3
Person responsible for the module	Dept. Head
Lecturer	Ms. Amel Zaougha
Language	English
Relation to curriculum	Professional Module, Compulsory
Type of teaching, contact hours	21 hours Seminar/ Project / Semester
Workload	Total 42 hours/ Semester (21 hours of Self Study)
Credit points	1.5
Requirements according to the examination regulations	-Minimum Attendance rate: 80% >20% of non-attendance= elimination for exams
Recommended prerequisites	English language skill (S1, S2)
Module objectives/intended learning outcomes	**Objectives:** 1. Learn how to write a clear and concise article that will appeal to a broad audience. **Learning Outcomes:** Students will be able to: 1. Write an abstract 2.Structure project data 3. Write a full publication (in Poster session)

Content	**Classroom Lecture** **Chapter 1. Elements of writing style** In this chapter student will learn: • some simple changes you can make when writing to make your papers easier to read • how to use different paragraph types and transitions to create focus and flow in your manuscript • how to improve your writing style through exercises. **Chapter 2. Titles and abstracts** In this chapter student will learn: • what makes a good title • two easy-to-use abstract templates that you can apply directly to your own writing • what to avoid when writing titles and abstracts. **Chapter 3. From introduction to conclusion** In this chapter student will learn: • what should be included in, or excluded from, each section of the paper • how to organize your ideas effectively and avoid common mistakes • how to create a narrative flow to help readers follow your argument. **Chapter 4. Data management** In this chapter student will learn: • why it is important to accurately record, share and preserve your data • the best approaches to managing your research data • the importance of sharing your data and who might use it. **Chapter 5. Data presentation** In this chapter student will learn: • the four principles of creating clear and engaging figures for your paper • how to choose which figures to include in your manuscript • best practices in raw data processing and image preparation • tips to ensure that your figures are clear and informative for your readers. **Project:** **Content: Each Student will prepare a publication for the Poster session organized every year at the end of the Annual Project.**

Study and examination requirements and forms of examination	Evaluation of Publication, Poster and oral presentation (100%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online (Moodle ULT) Video projection
Reading list	1. UC Berkeley Student Learning Center https://slc.berkeley.edu/writing-worksheets-and-other-writing-resources

U3.5 Languages and Management I

Project Management

Module designation	**Languages and Management I**
Module level, if applicable	Year 2, Semester 1
Code, if applicable	U3.5
Subtitle, if applicable	
Courses, if applicable	**Project Management**
Semester(s) in which the module is taught	**Semester 3**
Person responsible for the module	Dept Head
Lecturer	Ms. Yosra Saidi Chabene
Language	French
Relation to curriculum	Compulsory module
Type of teaching, contact hours	21 hours of Classroom Lecture/ Semester
Workload	Total 42 hours/ Semester (21hours of Self Study)
Credit points	1.5
Requirements according to the examination regulations	- Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	
Module objectives/intended learning outcomes	**The objectives** of this course are to: 1. To make them understand the concepts of Project Management for planning to execution of projects. 2. To make them understand the feasibility analysis in Project Management and network analysis tools for cost and time estimation. 3. To enable them to comprehend the fundamentals of Contract Administration, Costing and Budgeting. 4. Make them capable to analyze, apply and appreciate contemporary project management tools and methodologies in Indian context. **Course Outcomes** On completion of this course, the students will be able to: 1.Understand project characteristics and various stages of a project. 2. Understand the conceptual clarity about project organization and feasibility analyses – Market, Technical, Financial and Economic. 3. Analyze the learning and understand techniques for Project planning, scheduling and Execution Control. 4. Apply the risk management plan and analyse the role of stakeholders.

	5. Understand the contract management, Project Procurement, Service level Agreements and productivity. 6. Understand the How Subcontract Administration and Control are practiced in the Industry.
Content	1. Introduction to Applied Project Management 2. Project Definition: Project Feasibility Analysis 3. Developing a Project Execution Plan 4. Setting up a Project Organization 5. Resource Scheduling, Cost Estimating 6. Controlling Project Execution, Project Control: Planning and Scheduling 7. Cost Engineering and Detailed Engineering 8. Project Procurement 9. Construction Management, Construction Progress, Productivity and Supervision 10. Subcontract Administration and Control
Study and examination requirements and forms of examination	Written Mid-Term Exam (40%) + Written Final Exam (60%)
Media employed	Course Material (Hard/ Soft copy) for Classroom & Online(Moodle ULT) Video projection
Reading list	1. Chaudhary S.; Project Management, Tata Mc Graw Hill 2. Kerzner H.; Project Management, II Edition, CBS Publishers 3. Meredith Jack R., Mantel Samuel J.; Project Management, IV Edition, John Wiley & Sons 4. Gopalakrishnan P., Ramamoorthy V.E; Textbook of Project Management, MacMillan Publishers 5. Maylor Harvey, Project Management, MacMillan Publishers 7. Matheen A. Prof., Comprehensive Project Management, Laxmi Publications (P) Ltd.

U3.5 Soft skills & Languages I

English TOIEC I

Module designation	Languages and Management 1
Module level, if applicable	2nd year
Code, if applicable	U3.5
Subtitle, if applicable	
Courses, if applicable	English TOEIC 1
Semester (s) in which the module is taught	Semester 3 (S3)
Person responsible for the module	
Lecturer	Ms. Nadia Zardi
Language	English
Relation to curriculum	Integrated course module
Type of teaching, contact hours	Lecture, 21 hours of classroom course/ semester
Workload	Total 42 hours/semester (21 hours of Self-Study/semester)
Credit points	2 credits
Requirements according to the examination regulations	Minimum attendance rate: 80% of the total contact hours >20 % of nonattendance = elimination for exams
Recommended prerequisites	General English 1 & 2
Module objectives/intended learning outcomes	**Objectives:** 1. Pass the TOEIC test 2. Grasp most of all listening and reading tactics **Learning Outcomes:** Students will be able to: 1. Demonstrate how well they understand spoken English. 2. Focus on their learning, think actively, monitor their comprehension of different types of texts and demonstrate appropriate reading strategies.

Content	**Classroom Lecture** **Part I: Listening** **Chapter 1. Photographs** Distractor 1: Sound Confusion Distractor 2: Verb/Noun Confusion Distractor 3: Non-Itemed Pictures Distractor 4: Action /State confusion **Mini Test** **Chapter 2. Question and Response** Distractor 1: Repeating words Distractor 2: Related words Distractor 3: Wrong Subject Distractor 4: Wrong Tense Answering Wh-Questions with Yes or No Distractor 5: Negative Questions Distractor 6: Tag Questions **Mini Test** **Chapter 3. Conversations (two or more speakers)** Distractor 1: Topic Questions Distractor 2: Detail Questions Distractor 3: Inference Questions Distractor 4: Types of situations **Mini Test** **Chapter4. Talks (one single speaker)** Distractor 1: Topic Questions Distractor 2: Speaker/Audience Questions Distractor 3: Detail Questions Distractor 4: Types of Talks **Mini Test** **Part II. Reading** **Chapter 1. Incomplete Sentences** Vocabulary Based Items

	Grammar Based Items **Mini Test** **Chapter 2. Text Completion** Grammar Concepts **Mini Test** **Chapter 3. Reading Comprehension** Question Types Passage Types Multiple passage Items **Mini Test**
Study and examination requirements and forms of examination	Continuous assessment through mini test A common test for all types of Engineering by the end of each semester Listening Exam (Semester I): 100 questions /multiple choice: 45minutes. Listening and Reading Exam (Semester II): 200 questions/ multiple-choice 120 minutes.
Media employed	Textbook/ hard copies for Classroom
Reading list	1. Dooley, J. (2019). Prepare and Practice for the TOEIC Test. Express Publishing. 2. G rant, T. (2007). Tactics for TOEIC. Oxford University Press