PEOPLE'S DEMOCRATIC REPUBLIC OF ALGERIA

MINISTRY OF HIGHER EDUCATION AND SCIENTIFIC RESEARCH

NEW

SPECIALIZED ENGINEERING TRAINING COURSES

(FROM GRADE 3TH TO GRADE 5TH)

Establishment	Faculty	Department
Blida University1	Sciences	IT

Domain	Channel	Speciality
Mathematics and Computer Science	IT	Computer security

Academic year: 2024-2025.

CONTENTS

II- Semester organization sheet.

I - Training identity sheet.

1 - Training location.

Faculty: Science Department:
Computer Science

2- Training partners[1].

- **Other academic institutions** :

- **Companies and other socio-economic partners:**

- **International partners :**

3- Training context and objectives.

A- General training organization: Project position.

B- Training objectives.

As the world of IT is constantly evolving, the present training offer in IT Security has the following main objectives:

- Train engineers with advanced IT security skills, enabling them to protect information systems, networks and data against threats and attacks. Engineers will be able to design and implement robust security solutions to guarantee the confidentiality, integrity and availability of information.
- Through this training, students will acquire in-depth knowledge of cryptography, network security, web application security, security of

[1] Present the agreements in the appendix to the training course.

systems, and risk management. They will also be trained in the use of incident detection and response tools and techniques.

- Train skills to develop security policies and procedures for organizations, carry out security audits, and advise companies on IT security best practices. IT security engineers will also be able to manage security incidents, investigate security breaches and implement disaster recovery plans.
- Students will be taught in-depth academic concepts, enabling them to pursue post-graduate studies in computer security, cybercrime or related fields. This will prepare them to make a significant contribution to research and development in the field of computer security.

C- Target profiles and skills...

- Master the principles and techniques of computer security, including cryptography, network security and system security.
- Know how to assess and manage IT security risks, including the implementation of security policies and the management of security incidents.
- Master current IT security tools and technological environments (firewalls, intrusion detection systems, VPNs, etc.).
- Design and implement secure architectures for information systems.
- Master application security techniques, including penetration testing and security audits.
- Know how to manage IT security projects, from planning to execution, in compliance with industry standards and best practices.
- Be able to train users and make them aware of good IT security practices.
- Respond to companies' security needs, proposing appropriate solutions and keeping a technological watch to anticipate new threats.
- Master new technologies and their impact on the security of corporate information systems.
- Be able to work collaboratively on IT security projects, using agile methodologies and project management techniques.

D- Regional and national employability potential.

The benefits of this training will be felt in both the regional and national contexts, given the immense need for skills in the field of safety.

computer science (at all levels) for the public and private sectors. Employment opportunities are available for the following profiles:

- IT security engineer,
- IT security analyst,
- Cybersecurity consultant,
- IT security auditor,
- Expert in cryptography,
- Etc.

E- Gateways to other specialties.

Bridges can be made with other IT engineering specialties (Systems and Network Security, Cybersecurity, etc.), as current IT Security training includes fundamental credits in systems security that can be used as credits in other IT specialties.

F- Training follow-up indicators.

- Pedagogical committees,
- Periodic review meetings,
- Monitoring student placement in the business sector

G- Management skills.

The number of students we can cater for: 40 to 50.

II- Semester organization sheet.

Teaching units	VHS	VH Weekly				Coeff.	Cr
	14 Sem.	Courses	TD	TP	Work Staff		
Fundamental Teaching Unit (UEF)							
UEF51 :	**126h**	**3h**	**3h**	**3h**	**6h**	**8**	
Mathematical Tools for Cryptography		1h30	1h30	1h30	3h	4	
Operational Research		1h30	1h30	1h30	3h	4	
UEF52 :	**105h**	**3h**	**3h**	**1h30**	**6h**	**6**	
Compilation		1h30	1h30	1h30	3h	4	
Software Engineering		1h30	1h30		3h	2	
Methodological Teaching Unit (UEM)							
UEM5 :	**84h**	**3h**		**3h**	**6h**	**4**	
Python Programming		1h30		1h30	3h	2	
Web Development		1h30		1h30	3h	2	
Discovery Teaching Unit (UED)							
UED5 :	**42h**	**1h30**	**1h30**		**3h**	**1**	
Theory of Information and Coding		1h30	1h30		3h	1	
Transversal Teaching Unit (UET)							
UET5 :	**21h**	**1h30**			**3h**	**1**	
Business Intelligence		1h30			3h	1	
TotalS	**378h**	**12h**	**7h30**	**7h30**	**24h**	**20**	

enester5							

Teaching units	VHS	VH Weekly				Coeff.	Cr
	14 Sem.	Courses	TD	TP	Work Staff		
Fundamental Teaching Unit (UEF)							
UEF61 :	**126h**	**3h**	**3h**	**3h**	**6h**	**7**	
Advanced Cryptography		1h30	1h30	1h30	3h	4	
Modeling and Simulation		1h30	1h30	1h30	3h	3	
UEF62 :	**105h**	**3h**	**1h30**	**3h**	**6h**	**6**	
Cloud Computing		1h30		1h30	3h	3	
Advanced Databases		1h30	1h30	1h30	3h	3	
Methodological Teaching Unit (UEM)							
UEM6 :	**84h**	**3h**	**1h30**	**1h30**	**6h**	**5**	
Mobile Development		1h30		1h30	3h	2	
Digital Signal Processing		1h30	1h30	1h30	3h	3	
Discovery Teaching Unit (UED)							
UED6 :	**42h**	**1h30**	**1h30**		**3h**	**1**	
AI Notions and Principles		1h30	1h30		3h	1	
Transversal Teaching Unit (UET)							
UET6 :	**21h**	**1h30**			**3h**	**1**	
Startup and Professional Development		1h30			3h	1	
Total Seneste	**378h**	**12h**	**7h30**	**7h30**	**24h**	**20**	

r 6							

Teaching units	VHS	VH Weekly				Coeff.			
	14 Sem.	Courses	TD	TP	Work Staff			Continuous	Review
Fundamental Teaching Unit (UEF)									
UEF71 :	**126h**	**4h30**	**1h30**	**3h**	**3h**	**7**	**10**		
Advanced Operating Systems		1h30		1h30	3h	3	5	40%	60%
Advanced Networks		3h	1h30	1h30		4	5	50%	50%
UEF72 :	**105h**	**3h**		**4h30**	**6h**	**6**	**10**	40%	60%
Computer Systems Security		1h30		3h00	3h	3	5	40%	60%
Information and Data Security		1h30		1h30	3h	3	5	40%	60%
Methodological Teaching Unit (UEM)									
UEM7 :	**84h**	**3h**	**1h30**	**1h30**	**6h**	**4**	**7**		
Programming by Constraint		1h30	1h30		3h	2	3	40%	60%
Machine Learning, Deep Learning, and Security		1h30		1h30	3h	2	4	40%	60%
Discovery Teaching Unit (UED)									
UED7 :	**42h**	**1h30**		**1h30**	**3h**	**2**	**2**		
Malware Analysis		1h30		1h	3h	2	2	40%	60%

				30					
Transversal Teaching Unit (UET)									
UET7 :	**21h**	**1h30**			**1h30**	**1**	**1**		
Critical Thinking and Creativity Skills		1h30			1h30	1	1		100%
Total Semester 7	**378h**	**13h30**	**3h**	**10h30**	**19h30**	**20**	**30**		

Teaching units	**VHS**	**VH Weekly**				**Coeff.**	**Credits**		
	14 Sem.	**Courses**	**TD**	**TP**	**Work Staff**			**Continuous**	**Review**
Fundamental Teaching Unit (UEF)									
UEF81 :	**84h**	**3h**		**3h**	**6h**	**6**	**10**		
Operating Systems Security		1h30		1h30	3h	3	5	40%	60%
Cybersecurity		1h30		1h30	3h	3	5	40%	60%
UEF82 :	**126h**	**3h**	**1h30**	**4h30**	**6h**	**7**	**10**		
Network Security		1h30	1h30	1h30	3h	4	5	50%	50%
Wireless and Mobile Network Security		1h30		3h00	3h	3	5	40%	60%
Methodological Teaching Unit (UEM)									
EMU8:	**105h**	**3h**	**1h30**	**3h**	**6h**	**3**	**5**		
Identity & Access Management		1h30		1h30	3h	1	2	40%	60%
Secure Software Development		1h30	1h30	1h30	3h	2	3	50%	50%
Discovery Teaching Unit (UED)									
UED8 :	**21h**	**1h30**			**3h**	**1**	**1**		
Innovation and Entrepreneurship		1h30			3h	1	1		100%

Transversal Teaching Unit (UET)									
UET8:	**42h**			**3h**	**3h**	**3**	**4**		
Multidisciplinary Project				3h	3h	3	4	100% Defense	
Total Semester 8	**378h**	**10h30**	**3h**	**13h30**	**21h**	**20**	**30**		

Teaching units	VHS	VH Weekly				Coeff.			
	14 Sem.	Courses	TD	TP	Work Staff			Continuous	Review
Fundamental Teaching Unit (UEF)									
UEF9 :	**126**	**4h30**		**4h30**	**9h**	**11**	**18**		
Web and mobile application security		1h30		1h30	3h	4	6	40%	60%
Embedded Systems Security		1h30		1h30	3h	4	6	40%	60%
Digital Forensics		1h30		1h30	3h	3	6	40%	60%
Methodological Teaching Unit (UEM)									
UEM9 :	**84h**	**3h**		**3h**	**6h**	**5**	**8**		
DevOps		1h30		1h30	3h	3	5	40%	60%
Ethical Hacking		1h30		1h30	3h	2	3	40%	60%
Discovery Teaching Unit (UED)									
UED9 :	**63h**	**3h**	**1h30**		**3h**	**3**	**3**		
Project Management		1h30	1h30		3h	2	2	40%	60%
Emerging Security Technologies		1h30			3h	1	1		100%
Transversal Teaching Unit (UET)									
UET9:	**21h**	**1h30**			**3h**	**1**	**1**		
Academic Communication and Research		1h30			3h	1	1		100%
Total Semester 9	**294h**	**10h30**	**1h30**	**7h**	**21h**	**20**	**30**		

				30					

6- Semester 10

Domain Mathematics and Computer Science.
Branch Computer Science.
Specialization Computer Security.

Research topic or work placement, culminating in a dissertation and oral presentation. Subjects must be assigned at the beginning of the year (October).

	VHS	Coeff.	Credit
Personal work			
Internship			
Seminars	125h	06	10
Other (Memory)	250h	14	20
Total Semester 10	**375h**	**20**	**30**

7- Overall training summary.

EU VH	UEF	UEM	UED	UET	Total
Courses	420h	210h	84h	105h	819h
TD	189h	63h	42h	-	294h
TP	420h	168h	63h	-	651h
Personal work	756h	420h	147h	189h	1512h
Semester 10	250h	125h	-	-	375h
Total	**2035h**	**986h**	**336h**	**294h**	**3671h**
Credits	128	36	11	5	**180**
% in Credits for each EU	71.11%	20%	6.11	2.78	100%

III - Detailed program by subject

Subject title: Mathematical Tools for Cryptography.
Credits: 06.
Coefficient: 04.

Teaching aim : The first part introduces fundamental notions for group theory, notions useful for understanding bodies and linear codes as well as their applications. The second part should allow the student to acquire the elementary knowledge provided by the theory of finite bodies.

Recommended prior knowledge: Some algebra concepts.

Contents :

Part 1.

1. Groups, examples.
2. Homomorphisms.
3. Subgroups, distinguished subgroups and quotient groups.
4. Cyclic groups, order of elements, index of a subgroup.
5. Center, centralizer, conjugation.
6. Special groups.
7. Permutation groups, matrix groups.
8. Examples of applications in cryptography.

Part 2.

1. Definitions, characteristics, cardinality of a finite field.
2. Frobenuis relation, Frobenuis morphism.
3. Construction and uniqueness of finite bodies, practical construction of Fq.
4. Sub field of a finite field, primitive element, polynomial primitive.
5. Irreducible polynomials and conjugate elements.
6. Factorization of x^(n) -1)
7. Congruences and Residual Classes.
8. Euler's Phi function, the Theorems of Fermat, Euler and Lagrange.
9. Quadratic residue.
10. Recurrent sequences and shift register.
11. Application examples: cryptographic keys.

Assessment methods: Continuous evaluation, and exam.

References :

1. J. Querre, Cours d'algèbre, Maitrise de Mathématiques, Masson. 1976.
2. J. Calais. Elements of group theory. PUF, 1998.
3. E. Ramis, C. Deschamps, and J. Odoux. Cours de Mathématiques 1, Algèbre. Dunod, 1998.
4. D.J.S. Robinson, "A course in the Theory of Groups," 2nd ed, Springer-Verlag, New York, 1995.
5. Rudolf Lid land Harald Niederreiter, Finite fields, Encyclopedia of Mathematics and applications, Cambridge University press, 1997.
6. M. Demazure. Algebra course. Primality, divisibility, codes. Cassini, 1997.

Subject title: Operational Research.
Credits: 04.
Coefficient: 04.

Teaching aim : To introduce the student to problem representation, data gathering and providing answers.

Recommended prior knowledge: Basic notions of mathematics.

Contents :

I. **Optimization in operational research.**
- Model: represent a problem.
- Instantiate: gather data.
- Solve: provide an answer.
- Examination of some situations.

II. **Linear programming in continuous variables.**
- Formulations.
- Geometric properties.
- Simplex algorithms.
- Duality.
- Additional differences.

III. **Linear programming in integer and mixed variables.**
- Formulations.
- Relaxation.
- Easily solvable problems, total unimodularity.
- Branch and bound method.
- Situations mixing continuous variables and integer variables.
- Reference combinatorial optimization problem.

IV. **Local optimization.**
- Constraint-free optimization (optimal conditions, linear search methods, etc.).
- Optimization with constraint (optimal conditions, quadratic sequential programming, etc.).

V. **Graphs theory.**
- Become familiar with the basic terminology of graph theory.
- Discover how to represent graphs in computer memory.
- Examine and implement various graph traversal algorithms.
- Learn how to implement a shortest path algorithm.
- Examine and implement the minimum spanning tree algorithm.
- Explore topological sort.
- Learn how to find Euler circuits in a graph.

Assessment methods: Continuous evaluation, and exam.

References :

1. D. de Werra, and T. M Liebling, Operational Research for engineers, polytechnic presses, 2003.
2. J-M. Hélary, and R. Pédrono, Operational Research: Guided Work, Hermann, 1983.
3. Y.Nobert, and R.Ouellet, Operational Research (3rd edition), Gaëtan Morin, 2002.

Title of subject: Compilation.
Credits: 6.
Coefficient: 4.

Teaching aim : The student will be able to differentiate between compiler and interpreter, the different phases of compilation, until the generation of the final code.

Recommended prior knowledge: Programming, and Language Theory.

Contents :

I. Introduction to compilation.
- The different stages of compilation.
- Compilation, interpretation and translation.

II. Lexical analysis.
- Regular expressions.
- Grammars.
- Finite step automata.
- An example of a lexical analyzer generator: LEX.

III. Syntactic analysis.
- Definitions: Syntactic grammar, left recursion, left factorization, free grammar.
- Calculation of the sets of first and following.
- Descending analysis methods: Recursive descent, LL (1).
- Bottom-up analysis methods: LR (1), SLR (1), LALR (1), item method.
- An example of a parser generator: YACC.

IV. Syntax-driven translation.

V. Intermediate forms.
- Post fixed shape
- Quadruplets.
- Direct and indirect triplets.
- Abstract tree.

VI. Allocation - Substitution - Organization of data at runtime.

VII. Object Code Optimization.

VIII. Object Code Generation.

Assessment methods: Continuous evaluation, and exam.

References :

1. Alfred Aho, Ravi Sethi, Compilers: Principles, techniques and tools - Courses and exercises -, DUNOD 2000.
2. Benjamin Cummings, A Retargetable Compiler: Design and implementation, Addison Wesley 1995.

Subject title: Software Engineering.
Credits: 4.
Coefficient: 2.

Teaching aim : To learn how to apply an analysis and design methodology for software development. In particular, to learn object modelling using the universal UML language.

Recommended prior knowledge: Algorithms, Information Systems, Object-Oriented Programming.

Contents :

Chapter I. Introduction to Software Engineering.

1. Definitions and objectives.
2. Principles of Software Engineering.
3. Expected qualities of software.
4. Software life cycle.
5. Software lifecycle models.

Chapter II. Information System design methods.

1. The challenges of the systems approach.
2. Concept of a system.
3. Typology of systems.
4. System design methods.
   4.1 Static system design.
      4.1.1 STB.
      4.1.2 SADT method.
      4.1.3 Entity Association Model.
   4.2 Dynamic system design.
      4.2.1 Prototyping.
      4.2.2 Object-oriented approaches.

Chapter III. Modelling with UML.

1. Introduction (Modelling, Model, Object Oriented Modelling, UML in application.).
2. General elements and mechanisms.
3. UML views and diagrams.
4. Packages.

Chapter IV. UML: Functional view & Static view.

1. Use case diagram.
2. Class diagram.
3. Object diagram.

Chapter V. UML: Dynamic view.

1. Interaction diagram (Sequence and collaboration).
2. Activity diagram.
3. State/transition diagram.

Assessment methods: Continuous evaluation, and exam.

References :

1. Pierre Gérard, Software Engineering: Principles and Techniques. A course for Licence Pro, Université de Paris 13 LIPN. FC 2007/2008.
2. Yann-Gaël Guéhéneuc, Project management for software development and maintenance. Course at the Department of Computer Science and Operations Research, Université de Montréal, Canada, 2003.
3. Yende Raphael Grevisse, Support de cours en génie logiciel 2, Course taught at Institut Supérieur de Commerce en Deuxième Licence CSI, 2019.
4. Olivier Guibert, Information Systems Analysis and Design Course (Tools and Models for Software Engineering), Computer Science Department, IUT, Université Bordeaux 1. November, 2007.
5. Delphine Longuet, Introduction au génie logiciel et à la modélisation, Cours au Polytech Paris-Sud Formation initiale 3eme Année Spécialité Informatique Année 2017-2018.

Title of the subject : Python Programming.
Credits: 4.
Coefficient: 2.

Teaching aim : This course could be a self-study document for a Python programming course. It contains a section for beginners, a discussion of several advanced topics of interest to Python programmers.

Recommended prior knowledge: Algorithms, and Object-Oriented Programming.

Contents :

Part 1- Beginning Python

- Lexical matters.
- Statements and inspection -- preliminaries.
- Built-in data-types.
- Functions and Classes -- A Preview.
- Statements.
- Functions, Modules, Packages, and Debugging.
- Classes.
- Special Tasks.

Part 2- Advanced Python.

- Regular Expressions.
- Iterator Objects.
- Unit Tests.
- Extending and embedding Python.
- Parsing.
- GUI Applications.
- Guidance on Packages and Modules.
- End Matter.

Part 3- Python Workbook.

- Lexical Structures.
- Execution Model.
- Built-in Data Types.
- Statements.
- Functions.
- Object-oriented programming and classes.
- Additional and Advanced Topics.
- Applications and Recipes.

Part 4- Generating Python Bindings for XML.

- Generating the code.
- Using the generated code to parse and export an XML document.
- Some command line options you might want to know.
- The graphical front-end.
- Adding application specific behavior.
- Special situations and uses.
- Some hints.

Assessment methods: Continuous evaluation, and exam.

References :

1. https://www.davekuhlman.org/python_book_01.pdf.
2. Black Hat Python: Python Programming for Hackers and Pentesters, Justin Seitz.
3. The Practice of Network Security Monitoring: Understanding Incident Detection and Response, Richard Bejtlich.

Subject title: Web Development.
Credits: 3.
Coefficient: 2.

Teaching aim : Mastery of programming and development of applications and Websites.

Recommended prior knowledge: Algorithmics, and HMI.

Contents :

1- Introduction to the Web.
2- Web architecture.
3- Web sites.
4- Web applications.
5- Web design and development.

- HTML5.
- CSS3.
- PHP5.
- SQL
- JAVASCRIPT language.
- jQuery library.
- Other tools.

Assessment methods: Continuous evaluation, and exam.

References :

1. Francis Draillard, Premiers pas en CSS3 et HTML5, 7th updated edition, Eyrolles, 2017.
2. Patrick Lenormand, How to boost the content of your website? Edition PYRAMYD, Collection: Savoir et savoir-faire, 2017.
3. Luc Van Lancker, AJAX - Développez pour le Web 2.0, Entrez dans le code : JavaScript, XML, DOM, XML http Request 2 ... Eni editions, collection: Ressources informatiques. 2015.
4. Jean-Marie Defrance, jQuery-Ajax avec PHP: 44 workshops to master jQuery. Publisher: Eyrolles, 4th edition, Collection: Blanche, 2013.
5. Bogdan Brinzarea, CristianDarie, Audra Hendrix, AJAX and PHP: How to build responsive web applications, Dunod, 2nd edition, Collection: InfoPro - Etudes, développement et intégration, 2010.

Subject title: Theory of Information and Coding.
Credits: 2.
Coefficient: 1.

Teaching aim : The aims of this course are to introduce the principles and applications of information theory. The course will study how information is measured in terms of probability and entropy, and the relationships among conditional and joint entropies; how these are used to calculate the capacity of a communication channel, with and without noise; coding schemes, including error correcting codes; how discrete channels and measures of information generalize to their continuous forms; the Fourier perspective; and extensions to wavelets, complexity, compression, and efficient coding of audio-visual information.

Recommended prior knowledge: Basics notions on coding information.

Contents :

1. Entropy and information, conditional entropy, mutual information.
2. Source coding: Huffman coding, Lempel-Ziv compression.
3. Channel coding.
4. Error Correcting codes, linear codes.
5. Code terminals and wire-tap channels.
6. The main families of block codes.
7. Decoding.
8. Cryptography and cryptanalysis.

Assessment methods: Continuous Evaluation, and Exam.

References :

1. William Cary Huffman, and Vera Pless, Fundamentals of Error-Correcting Codes, Cambridge University Press, 2010.
2. David JC MacKay. Information Theory, Inference, and Learning Algorithms, 2003.
3. Olivier Rioul, Information and coding theory, 2007.

UE : UET5.
Subject title: Business Intelligence.
Credits: 1.
Coefficient: 1.

Recommended prior knowledge: Business notions.

Content: Business Intelligence (BI) is a crucial aspect of modern business strategy, focusing on the utilization of data-driven insights to make informed decisions and gain a competitive advantage. This course introduces students to the concepts, technologies, and practices of BI, covering topics such as data warehousing, data mining, analytics, visualization, and decision support systems. Through lectures, case studies, and hands-on projects, students will learn how to collect, analyze, and interpret data to support organizational decision-making and improve business performance.

Contents :

- Introduction to Business Intelligence.
- Data Warehousing and ETL Processes.
- Data Modeling and Dimensional Design.
- Data Mining and Predictive Analytics.
- BI Tools and Technologies.
- Advanced Analytics and Big Data.
- Business Intelligence Applications and Case Studies.

Assessment method: Exam.

References :

1. MÜLLER, Roland M. and LENZ, Hans-Joachim. Business intelligence. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.
2. HOWSON, Cindi. Successful business intelligence. Emeryville: McGraw-Hill Professional Publishing, 2007.
3. MICHALEWICZ, Zbigniew, SCHMIDT, Martin, MICHALEWICZ, Matthew, et al. Adaptive business intelligence. Springer Berlin Heidelberg, 2006.

UE : UEF61.
Subject title: Advanced Cryptography.
Credits: 6.
Coefficient: 4.

Teaching aim : Introduce the student to the study of cryptosystems based on algebraic problems or error-correcting code problems.

Recommended prior knowledge: Some algebra concepts.

Contents :

1. Introduction.
   - a- Security needs.
   - b- Symmetric Crypto-Systems, Asymmetric Crypto-Systems.
   - c- Hash Functions.
   - d- Electronic Signature.
   - e- New Trends in Cryptography.
   - f- Cryptanalysis.
2. Encryption, security.
   - a- "One-way function.
   - b- The RSA method and factorization of integers.
   - c- Discrete logarithm and El Gamel cryptosystem.
   - d- The Knapsack problem.
   - e- Error correcting codes and Mc Elièce cryptosystem.
   - f- Elliptic curves, cryptosystems.
   - g- Secret Sharing.
   - h- Image encryption.
   - i- Copyright protection.
3. Authentication.
   - a- Protocols, Principles.
   - b- Authentication techniques, digital signature.
   - c- Signature using public keys.
   - d- File security.
   - e- Algorithms, examples.

Assessment methods: Continuous Evaluation, and Exam.

References :

1. Ireland & Rosen, A Classical Introduction to Modern Number Theory, Springer.
2. Koblitz, A Course in Number Theory and Cryptography, Springer, 1994.
3. Blake, Seroussi and Smart, Elliptic Curves in Cryptography, Springer.
4. Koblitz, Algebraic Aspects of Cryptography, Springer.

Specialty: Computer Security.
Semester: 06.
UE : UEF61.
Topic title: Modeling & Simulation.
Credits: 6.
Coefficient: 3.

Teaching aim : This course is intended to deepen the knowledge of the student in modeling and simulation field. In addition, it introduces techniques of performance evaluation.

Recommended prerequisites: Engineering science, Mathematics, Automation.

Contents :

I. Systems modeling.
I.1 Definitions.
  I.1.1 Definition of modeling.
  I.1.2 System definition.
  I.1.3 Model definition.
I.2 Types of systems: discrete, continuous, deterministic.
I.3 Types of models: descriptive, analytical.
I.4 Modeling tools.
  I.4.1 Transfer function.
    I.4.1.1 Definition.
    I.4.1.2 Laplace transform.
    I.4.1.3 Block diagrams.
  I.4.2 Finite state machines.
  I.4.3 Petri net.
  I.4.4 Markov chains.
  I.4.5 Tail models.
II. Performance evaluation techniques.
  II.1 Presentation of techniques.
  II.2 Mathematical methods.
  II.3 Introduction to simulation.
III. Simulation.
  III.1 Types of simulation.
    III.1.1 Simulation of dynamic systems.
    III.1.2 Continuous simulation.
    III.1.3 Simulation of discrete systems.
  III.2 Sampling.
  III.3 Generation of pseudo-random numbers.
  III.4 Random number generator tests.
  III.5 Analysis and validation of simulation results.
IV. Simulation tools.
  IV.1 Software.
  IV.2 Languages.
  IV.3 Graphics and simulation.
V. Study of a simulation language.

Assessment methods: Continuous evaluation, and exam.

References :

1. Youssef Monsef, Modélisation et simulation des systèmes complexes : Concepts, methods and tools, Tec & Doc Lavoisier, 1996.
2. Frédéric Amblard, Denis Phan, Modélisation et simulation multi-agents Hermès, Science Publications, 2006.
3. J. Christian Attiogbé, Modélisation et construction des applications réparties, Modélisation avec les Réseaux de Petri, DUT Informatique - Module M-4102C, Janvier 2020.
4. Christophe Sabot, Part III: Markov chains: Informal lecture notes, Université Lyon-1, 2020.
5. Yliès Falcone, Jean-Claude Fernandez, Automates à états finis et langages réguliers, Book, Dunod, 2020.
6. Sara Rachidi, Fault diagnosis in discrete-event systems subject to temporal constraints, Thesis, Normandie Université, 2019.
7. S. Le Digabel, Introduction aux queues, Support de cours, Ecole Polytechnique de Montréal, 2017.

Specialization: Computer Security.
Semester: 06.
UE : UEF62.
Subject title: Cloud Computing.
Credits: 4.
Coefficient: 3.

Teaching aim : To allow the student to become familiar with the Cloud Computing, by presenting the foundations of virtualization as well as the tools to create and deploy Cloud infrastructures.

Recommended prior knowledge: Notions of virtualization, distributiveness, network, Web, ...

Contents :

Chapter I. Definitions and History.
I.1. Definitions.
I.1.1. The Cloud, and the Cloud Computing.
I.1.2. Cloud Computing from an Economic Viewpoint.
I.1.3. The Cloud Computing: A Virtual Space.
I.2. Historic.
I.2.1. The 50's.
I.2.2. Early 2000s.

Chapter II. Cloud Computing Models and Services.
II.1. Cloud Model.
II.2. Cloud Services.
II.2.1. Infrastructure as a Service: IaaS.
II.2.2. Platform as a Service: PaaS.
II.2.3. Software as a Service: SaaS.
II.2.4. Cloud Services Architecture.
II.2.5. Other Services.

Chapter III. Architecture and Typology of Cloud Computing.
III.1. Architecture.
III.1.1. N-Tiers.
III.1.2. Service Oriented Architecture (SOA).
III.1.3. Virtual Machine.
III.1.4. File Virtualization.
III.2. Deployment.
III.2.1. Pilot Phase.
III.2.2. Deployment and Integration Phase.
III.2.3. Loading Driving Phase.
III.3. Typology.
III.3.1. Private Cloud.
III.3.2. Public Cloud.
III.3.3. Community Cloud.
III.3.4. Hybrid Cloud.
III.3.5. Distributed Cloud.
III.3.6. Inter Cloud.
III.3.7. Multi Cloud.

Chapter IV. Cloud Examples.
IV.1. DROPBOX.
IV.2. Microsoft Cloud Platform.
IV.3. Commercials Clouds, and Main Market Players.
IV.5. OpenStack Overview.
IV.6. Examples of Cloud for Storage.

Chapter V. Benefits and Limits of the Cloud.
V.1. Benefits of the Cloud.
V.1.1. Cost Reduction.
V.1.2. Flexibility.
V.1.3. Refocusing on the Core Business.
V.2. Cloud Limitation.
V.2.1. Control Loss of Your IT (Entrusted to One or Third Parties).
V.2.2. Problems with Securing its Computer Data.
Chapter VI. Security and Privacy in the Cloud.
VI.1. General Aspects.
VI.2. Specific Security Issues.
VI.3. Contractual Aspects.
IV.4. Best Security Practices.
IV.5. Synthesis and Overview.
IV.5.1. Threat.
IV.5.2. Attacker Types.
IV.5.3. Security Risks.
IV.5.4. Advice for Limiting Risks.

Assessment methods: Continuous evaluation, and exam.

References :

1. Rajkumar Buyya, James Broberg, Andrzej M. Goscinski, "Cloud Computing: Principles and Paradigms", John Wiley & Sons, 2010 (ISBN 9781118002209).
2. Lee Gillam, "Cloud computing", Springer, 2010 (ISBN 9781849962414).
3. Zaigham Mahmood, Richard Hill, "Cloud Computing for Enterprise Architectures", Springer, 2011 (ISBN 9781447122364).
4. Cigref Réseau des grandes entreprises, "Fondamentaux du Cloud Computing - Le point de vue des grandes entreprises", March 2013.
5. Romain Hennion, Hubert Tournier, Eric Bourgeois, Cloud Computing: Décider - Concevoir - Piloter - Améliorer, Eyrolles, 2012.
6. Guillaume Plouin, Cloud Computing, Sécurité, stratégie d'entreprise et panorama du marché, Collection InfoPro, Dunod, 2013.
7. Guillaume Plouin, Tout sur le Cloud Personnel, Work, store, play and exchange... in the cloud, Dunod, 2013.

UE : UEF62.
Subject title: Advanced Databases.
Credits: 4.
Coefficient: 3.

Teaching aim : Follows the evolution of the IT context and the advent of system applications in existing databases while showing current trends. The course will also deal with database security.

Recommended prerequisites: Concepts on Database, DBMS

Contents :

1. Extended relational model
2. Semantic models (SDM, AI, etc.)
3. Object-oriented databases
4. Deductive databases
5. Distributed databases
6. Multimedia databases
7. Secure Databases and database security

Assessment methods: Continuous evaluation, and exam.

Subject title: Mobile Development.
Credits: 3.
Coefficient: 2.

Teaching aim : The student will acquire knowledge of application development in mobile environments. They are omnipresent whether you are a customer (B2C), supplier (B2B) or employee (B2E). He will learn programming under Android, its development platform and the specificities of embedded development on smartphones.

Recommended prior knowledge: Web development.

Contents :
Chapter 1: Mobile applications.
- Mobile Operating Systems.
- Mobile Application Types.

Chapter 2: Android Platform.
- Presentation of the Android platform.
- The fundamental components of an Android application.
- Android SDK.
- Installation and configuration of tools.
- Create an Android Emulator.

Chapter 3: Activities and resources.
- Concept of Activity.
- Life cycle of an activity.
- Resources, Organization of resources, and utilization.

Chapter 4: GUIs and Widgets.
- Creation of graphical interfaces.
- Manage events on widgets.

Chapter 4: Menus and dialog boxes.
- Management of application menus, Options menu, and Context menus.
- Dialog boxes.

Chapter 4: Communication between components: Explicit intents, Implicit intents, and Resolving Implicit Intents.
Chapter 5: Databases with SQLite. Chapter
6: Development of an application.

Assessment methods: Continuous evaluation, and exam.

Subject title: Digital Signal Processing.
Credits: 4.
Coefficient: 3.

Teaching aim : This course introduces the basic concepts and principles underlying continuous and discrete-time signal processing. The objective is to analyze, manipulate, and interpret signals to extract useful information or enhance their quality for various applications. The Concepts will be illustrated using examples of standard technologies and algorithms.

Recommended prior knowledge: Signal theory, applied mathematics.

Contents :

Chapter I. Introduction to Signal Processing.

1. Signal and System.
2. Signal Classification.
3. Frequency and Time Representation.

Chapter II. Analog Signal Processing.

1. Fourier Series.
2. Fourier Transform.
3. Convolution.
4. Filtering Concept.
5. Modulation Concept.

Chapter III. Digital Signal Processing.

1. Sampling.
2. Quantization.
3. Coding.
4. The Discrete Fourier Transform (DFT).
5. Discrete Fourier Transform: Derivation of Radix-2 FFT.

Chapter IV. Fast Algorithms for Signal Processing.

1. Fast Convolution Algorithm.
2. Fast Fourier Transform Algorithm.
3. Multidimensional Transform Algorithms.
4. Algorithms Derived from the Fourier Transform.

Chapter V. Wavelet Transform and Time-Frequency Analysis.

1. Multiresolution Analysis, Splines, and Wavelets.
2. Orthogonal Decomposition of Wavelet Series.
3. Wavelet Decompositions and Reconstructions.

Assessment methods: Continuous evaluation, and exam.

Title of the subject : AI: Notions & Principles.
Credits: 2.
Coefficient: 1.

Teaching objectives : Acquisition of fundamental and preliminary notions about AI.

Recommended prior knowledge: Difference between natural and artificial intelligence.

Contents :

Chapter 1: Birth of AI.

1- History: birth of AI, type of problem that AI addresses, and difference compared to computational computing.
2- Turing test.
3- Field of application of AI.

Chapter 2: Expert system.

1- Role definition.
2- Architecture of an OS.

Chapter 3: Operation of expert systems.

1- Notion of knowledge and representation formalism.
2- Production rules.
3- Operation of an inference engine.

Chapter 4: Approach to developing an expert system.

1. Expert system development process.
2. Example of an expert system: Dendral, Mycin, Prospector, etc.

Assessment method: Exam.

UE : UET6.
Subject title: Startup and Professional Development.
Credits: 1.
Coefficient: 1.

Teaching objectives :
- Understand the principles of entrepreneurship and startup development.
- Develop skills in idea generation, validation, and business model canvas creation.
- Learn effective pitching techniques and strategies for attracting investors.
- Gain insights into startup funding options and the venture capital landscape.
- Master professional development skills tailored for computer science students, including resume writing, networking, and job searching.
- Prepare for technical interviews and learn best practices for securing internships and full time positions in the tech industry.
- Explore avenues for career advancement and personal growth within the tech sector.

Recommended prior knowledge: Advanced business concepts.

Contents :

1. Introduction to Startups.
2. Idea Generation and Validation.
3. Business Model Canvas.
4. Pitching and presenting.
5. Startup Funding.
6. Professional Development for Computer Science Students.
7. Job Searching Strategies.
8. Interview Preparation.
9. Internships and Co-op Programs.
10. Career Advancement in Tech.

Assessment method: Exam.

UE : UEF71.
Subject title: Advanced Operating Systems.
Credits: 5.
Coefficient: 3.

Teaching aim : The objective of this course is to provide an in-depth study of the problems encountered in systems centralized and distributed operating systems. The basic mechanisms proposed for the resolution of the parallelism, mutual exclusion, synchronization, inter-process communication and deadlock are studied in detail. Directed and practical work allows students to manipulate and master the use of the basic mechanisms of the operating systems studied theoretically.

Recommended prerequisites: basic notions of operating systems, algorithms, machine structure, and the mechanisms allowing the management of machine resources, in particular the processor and memory.

Contents :

Chapter 1: Notions of parallelism, cooperation and competition.

a. Sequential processes.
b. Concept of task.
c. Task systems and precedence graph.
d. Task system language.
e. Task system state.
f. Determinism and maximal parallelism.
g. Cooperation and competition.
h. Thread concept.

Chapter 2: Synchronization between processes.

a. Mutual exclusion problem.
b. Implementation of mutual exclusion (lock, alternation, Peterson, TSL, sleep primitives, and wakeup).
c. Synchronization problem.
d. Implementing synchronization (event counters, semaphores, monitors).

Chapter 3: Inter-process communication.

a. Problematic.
b. Exchange of messages.
c. Mail boxes.
d. Communication tubes under Unix.
e. Signals.
f. Sharing variables (variables, files, data segments).
4. Chapter 4: Deadlock.
a. Introduction.
b. Deadlock.
c. Necessary conditions for deadlock.
d. Solutions to the deadlock problem.
e. Detection and recovery.
f. Deadlocks avoidance.
g. Prevention of deadlocks: PERSONAL WORK.

Assessment methods: Continuous evaluation, and exam.

Specialization: Computer Security.
Semester: 07.
UE : UEF71.
Subject title: Advanced Networks.
Credits: 5.
Coefficient: 4.

Teaching aim : This module aims to provide students with an in-depth understanding of advanced concepts, protocols, and technologies in computer networks. It builds upon foundational knowledge in networking and explores topics such as network security, emerging technologies, and advanced network architectures. Through lectures, practical exercises, and case studies, students will develop the skills and expertise necessary to design, implement, and manage complex computer networks.

Recommended prerequisites: Students should have a solid understanding of basic networking concepts, protocols, and technologies, as well as proficiency in network configuration and troubleshooting. Prior knowledge of programming languages, particularly Python or similar scripting languages, may be beneficial for certain topics such as network automation and SDN.

Contents :

1. Introduction to Advanced Computer Networks.
   - o Overview of the module objectives, structure, and assessment criteria.
   - o Review of fundamental networking concepts and protocols.
   - o Introduction to advanced networking topics and their relevance in modern network environments.
2. Network Security.
   - o Threats, vulnerabilities, and attacks in computer networks.
   - o Cryptography and encryption techniques for securing data transmission.
   - o Firewalls, intrusion detection systems, and other security mechanisms.
   - o Secure network design principles and best practices.
3. Quality of Service (QoS).
   - o Overview of QoS requirements and challenges in modern networks.
   - o Traffic shaping, prioritization, and scheduling techniques.
   - o QoS mechanisms in different network architectures, such as DiffServ and MPLS.
   - o Case studies and practical exercises on QoS implementation.
4. Emerging Network Technologies.
   - o Introduction to emerging technologies such as Software-Defined Networking (SDN), Network Function Virtualization (NFV), and Internet of Things (IoT).
   - o Overview of their architecture, protocols, and applications.
   - o Case studies and practical demonstrations of emerging network technologies.
5. Advanced Routing and Switching.
   - o Routing protocols beyond basic routing algorithms (e.g., OSPF, BGP).
   - o Advanced switching techniques and protocols (e.g., VLANs, Spanning Tree Protocol).
   - o Scalability, resilience, and performance considerations in routing and switching.
6. Network Management and Monitoring.
   - o Network management frameworks and protocols (e.g., SNMP, NetFlow).
   - o Configuration management, fault detection, and performance monitoring.
   - o Network troubleshooting methodologies and tools.
7. Wireless and Mobile Networks.
   - o Overview of wireless communication principles and technologies.
   - o Mobile network architectures (e.g., 4G/5G) and protocols (e.g., GSM, LTE).
   - o Security, QoS, and mobility management in wireless and mobile networks.

8. Case Studies and Practical Applications.
   - Real-world case studies of advanced network deployments and implementations.
   - Hands-on lab sessions and practical exercises to reinforce theoretical concepts.
   - Project work or assignments focusing on designing and implementing advanced network solutions.
9. Future Trends and Challenges.
   - Exploration of future trends and developments in computer networks.
   - Discussion of emerging technologies, challenges, and opportunities.
   - Ethical, legal, and societal implications of advanced network technologies.

Assessment methods: Continuous evaluation, and exam.

Specialty: Computer Security.
Semester: 07.
UE : UEF72.
Subject title: Computer Systems Security.
Credits: 5.
Coefficient: 3.

Teaching aim : One of the main objectives of this course is adversarial thinking: students should be able to quickly zoom in on the weakest link in any security technology, or system design. Students should be able to imagine how an attacker might break their system, and build in protection and mitigation measures to ward off such attacks.

Recommended prerequisites: Concept of computer security.

Contents :
- Principles and practice of building and administering secure systems.
- Authentication and access control.
- Operating system security.
- Program security.
- Key management.
- Information flow.
- Insurance.
- Vulnerability analysis and intrusion detection.

Assessment methods: Continuous evaluation, and exam.

UE : UEF72.
Subject title: Information and Data Security.
Credits: 5.
Coefficient: 3.

Teaching aim : This course allows students to acquire skills to ensure the security and proper functioning of computer systems.

Recommended prior knowledge: Algorithmic foundation, programming technique.

Contents :

I.1. Definitions: Security, Dependability, etc.
I.2. Main information security concepts: Vulnerability, threat, countermeasure, risk, ...
I.3. Information security objectives: Confidentiality, Integrity, Availability, Non-repudiation, Authentication.
I.4. Security types.
I.5. Security flaws.
I.6. Risk management process.
I.7. Risks typology and proposed solutions.
I.8. IT threats.

- What is an attack?
- Attacks motivations.
- Origin of attacks.
- Who can be targeted?
- Stages of an attack.
- Different taxonomies of attacks.
- Different types of attacks: Network attacks, System attacks, Password attacks, Website attack, application attack.
- Ways to launch an attack.
- Flaws and attacks (IP Spoofing, DoS, phishing, ...).
- Malware: Virus, Worm, Trojan horse, Spyware, …

I.9. Defense methods: Anti-virus, Firewalls, Private networks, Intrusion detection, etc...

Assessment methods: Continuous evaluation, and exam.

Topic title: Programming by Constraints.
Credits: 3.
Coefficient: 2.

Teaching objectives: Allow the student to use constraint programming techniques for solving complex combinatorial problems from logic programming and artificial intelligence.

Recommended prior knowledge: Problem solving in artificial intelligence using logic programming, Combinatorial Optimization.

Contents :

1- General information on Constraint Programming (CP).
- Introduction to Constraint Programming.
- Definition and fundamental principles of CP.
- Applications of CP in various fields.
- Examples of problems solved by CP.

2- Constraint Modeling.
- Representation of variables and constraints.
- Types of constraints and their properties.
- Global constraints and local constraints.
- Modeling techniques for specific problems.
- Modeling in Constraint Satisfaction Problem (CSP).
- Card Colorability, Magic Square, Golomb Rule, the n Queens, Euler's knight.
- Binarization.
  Binary CSP, Boolean CSP, Binary CSPs, n-ary CSPs.

3- Resolution Methods in CP.
- Systematic research techniques (backtracking, branch and bound).
- Constraint propagation and filtering algorithms.
- Heuristic search strategies (variable ordering, value ordering).

4- Practical Applications of PPC.

Assessment methods: Continuous evaluation, and exam.

Subject title: Machine Learning, Deep Learning, and Security.
Credits: 4.
Coefficient: 2.

Teaching aim : Understand ML and DL techniques and apply them to resolve security issues.

Subject title: Malware Analysis.
Credits: 2.
Coefficient: 2.

Teaching aim : This course aims to provide students with an in-depth understanding of malware and its attack techniques and acquire advanced skills in malware analysis by combining static, dynamic and behavioral approaches with the hybrid method and reverse engineering. It also prepares students to identify, analyze and neutralize complex and emerging malware.

Recommended prerequisites: Concepts on IT security risks and vulnerabilities.

UE : UET7.
Subject title: Critical Thinking and Creativity Skills.
Credits: 1.
Coefficient: 1.

Teaching objectives :
The aim of this course is to introduce the concept of critical thinking and its importance as well as give students the tools necessary to develop their critical thinking abilities and creativity skills.

Recommended prior knowledge: None.

Contents :

1. Introduction: Importance of skills for future employment. Importance of creativity and critical thinking as soft skills.
2. Introduction to analytical thinking, identifying, and evaluating arguments.
3. Various problem-solving methodologies.
4. Decision-making processes and risk analysis
5. Understanding logical fallacies and avoiding them in decision-making.
6. Exploring creativity, fostering a creative mindset, mind mapping and brainstorming, convergent and divergent thinking.
7. Critical Thinking in Coding: debugging and code review.
8. Integrating critical thinking and creativity for effective problem-solving.
9. Final Project and Presentation: students will integrate what they learned in a final project.

Assessment method: Exam (Final project presentation).

UE : UEF81.
Subject title: Operating Systems Security.
Credits: 5.
Coefficient: 3.

Teaching aim : The objective of this course is to allow the student to master the security of operating systems: the basic concepts, methods of analysis and evaluation of the security of operating systems (desktop and mobile). The student will learn about issues related to authentication, access control, and control flow integrity.

Recommended prerequisites: Full operating systems and the basics of computer security.

Contents :

1 - Introduction to operating system security (Linux, Windows, and Android).
2- Introduction to operating system administration and access control (Linux, Windows, and Android).
3 - Attacks on Oss.
4 - Operating system protection mechanisms.
5 - Methods for analyzing and evaluating the security of an operating system.
6 - Failure recovery and recovery methods.

Specialization: Computer Security.
Semester: 08.
UE : UEF81.
Subject title: Cybersecurity.
Credits: 5.
Coefficient: 3.

Teaching aim : The objectives of this course are to raise awareness about the importance of cybersecurity in today's digital world particularly in the context of businesses, equipping the students with foundational knowledge to protect themselves online, fostering a culture of security and responsibility, preparing them to comply with cybersecurity regulations and standards, and supporting their professional development in cybersecurity-related fields.

Recommended prior knowledge: Basic concepts of computer security and digital vulnerability analysis.

Assessment methods: Continuous evaluation, and exam.

Specialty: Computer Security.
Semester: 08.
UE : UEF82.
Subject title: Network Security.
Credits: 5.
Coefficient: 4.

Teaching aim : This course aims to provide students with an in-depth understanding of securing networks, regardless of their type or architecture. Primary objectives include the ability to identify best practices, tools and methodologies for analyzing and evaluating network security, as well as the design and implementation of secure network architectures.

Recommended prior knowledge: Basic concepts of computer security and digital vulnerability analysis.

Assessment methods: Continuous evaluation, and exam.

Specialization: Computer Security.
Semester: 08.
UE : UEF82.
Subject title: Wireless and Mobile Network Security.
Credits: 5.
Coefficient: 3.

Teaching aim : This course aims to provide students with an in-depth understanding of securing networks, regardless of their type or architecture. Primary objectives include the ability to identify best practices, tools and methodologies for analyzing and evaluating network security, as well as the design and implementation of secure network architectures.

Recommended prior knowledge: Basic concepts of computer security and digital vulnerability analysis.

Assessment methods: Continuous evaluation, and exam.

Subject title: Identity & Access Management.
Credits: 2.
Coefficient: 1.

Teaching aim : Identity and access management encompasses the tools and processes that are used to verify the identity of users and employees, authorize their access to defined resources (applications, tools, data), and monitor their actions.

Recommended prior knowledge: Basic concepts on identification and access rules.

Assessment methods: Continuous evaluation, and exam.

Subject title: Secure Software Development.
Credits: 3.
Coefficient: 2.

Teaching aim : This course aims to provide students with an in-depth understanding of the fundamental principles of software security from the design phase. It also allows learners to acquire skills in integrating security throughout the software development lifecycle (DevSecOps). At the end of this subject, the student is expected to be able to implement management practices effective security in software development, deployment and maintenance.

Recommended prerequisites: Software development approaches and basic notions of IT security.

Assessment methods: Continuous evaluation, and exam.

Subject title: Innovation and Entrepreneurship.
Credits: 1.
Coefficient: 1.

Teaching aim : The aim of this course is to motivate students to join the entrepreneurship world especially through the creation of viable economic and social solutions through small businesses, patents, or Startups. It continues from its predecessor.

Recommended prerequisites: Entrepreneurship basics from previous course.

UE : UET8.
Subject title: Multidisciplinary Project.
Credits: 4.
Coefficient: 3.

Teaching aim : The aim of this subject is the immersion of students in the socio-economic environment by placing them in internships in companies. The project takes place during the second semester of the fourth year. It consists of the design and carrying out a small IT project which takes place in a company.

Recommended background: Everything studied during the four years.

Project progress :

The project is described through precise specifications and can cover a wide variety of themes. It is proposed and supervised by a teacher from the department and must cover at least two disciplines.

The project group must be composed of 4 to 6 students. In addition to the technical content, which will consist of the application of the knowledge acquired for the implementation of the software development cycle, emphasis will be placed on the acquisition and application of organizational and relational aspects between the members of the group, the supervisor and the host company, respecting the following points:

- Analysis and division of work,
- Distribution of workloads between group members by the supervisor.
- Circulation of information between group members,
- Setting up a work schedule,
- Periodic presentations of project progress,
- Delivery of the final products set out in the project sheet,
- Writing an internship report (between 20 and 30 pages),
- Presentation of the work carried out before an examination committee.

Project evaluation :

The project evaluation will take the form of a score out of twenty and is based on the following criteria:

- The group submits an internship report and the software accompanied by a letter of presence in the host company.
- An examination committee composed of the supervisor, a teacher from the department and possibly a representative of the host company will examine the file in the presence of the group of students.
- The final grade is delivered to each student in the group (overall grade awarded to the team or individual in the event that it is noted that the volume of work provided by the members is unequal) according to the following scale:
- The internship report is graded on 6 points.
- The software is rated on 6 points.
- The presentation and the answers to the questions are marked out of 6 points.

(The mark awarded out of 18 is equal to the average of the marks awarded by the examination committee members).

- A continuous work mark (on 2 points) is given by the supervisor. This note will in some way validate the students' attendance at periodic meetings and compliance with the set objectives.

Assessment method: Exam (Project Defense).

UE : UEF9.
Subject title: Web and mobile application security.
Credits: 6.
Coefficient: 4.

Teaching aim : Understand the fundamental principles and concepts of secure Web browsing, Web development architecture, the main vulnerabilities and dedicated attacks on the Web, the mechanisms and best practices for developing and configuring Web applications. Understand the role of encryption in mobile application and device security and describe common scenarios in which processes encryption is applied.

Recommended prerequisites: Network Security, and Digital Vulnerability Analysis.

Specialization: Computer Security.
Semester: 09.
UE : UEF9.
Subject title: Embedded Systems Security.
Credits: 6.
Coefficient: 4.

Teaching aim : This subject aims to present the basic concepts of embedded systems, their security, and their specificities: reduced memory size, the need to process certain information in real time, the need to discover and control new peripherals. This subject also targets the programming of microcontrollers.

Recommended prior knowledge: Computer architectures and machine structure.

Assessment methods: Continuous evaluation, and exam.

UE : UEF9.
Subject title: Digital Forensics.
Credits: 6.
Coefficient: 3.

Teaching aim : This course provides students with an understanding of the fundamental process of analyzing data collected from electronic devices (including computers, media, and other digital evidence). Students will become familiar with the appropriate techniques and tools used to secure, manipulate, and preserve digital and multimedia evidence at physical crime scenes.

Recommended prior knowledge: IT systems, IT networking.

Assessment methods: Continuous evaluation, and exam.

Subject title: DevOps.
Credits: 5.
Coefficient: 3.

Teaching objectives: Allow the student to become familiar with the concepts of software project management with DevOps, as well as its tools.

Recommended prerequisites: Fundamentals of software engineering and Cloud Computing.

Assessment methods: Continuous evaluation, and exam.

Topic title: Ethical Hacking.
Credits: 3.
Coefficient: 2.

Teaching aim : This course introduces students to the fundamentals of ethical hacking, with a focus on understanding security vulnerabilities, performing penetration tests, and implementing countermeasures. Through a combination of theoretical courses and practical exercises, students will develop the skills necessary to identify and mitigate security threats within information systems.

Recommended prerequisites: Introduction to Operating Systems and Cybersecurity.

Subject title: Project Management.
Credits: 2.
Coefficient: 2.

Teaching objectives: Allow the student to understand the major issues of project management. Introduce the student to the process of organization and planning. Train the student in the application of planning processes, methods and tools. Introduce the student to project management environments.

Recommended prior knowledge: Project Notions.

Assessment methods: Continuous evaluation, and exam.

Subject title: Emerging Security Technologies.
Credits: 1.
Coefficient: 1.

Teaching aim : This course will cover topics like blockchain, cryptocurrency and quantum computing and other novel tech in cybersecurity. Its main goal is, thus, to explore the cutting-edge technologies of the cybersecurity world.

Recommended prerequisites: Computer Security Concepts.

Assessment method: Exam.

Specialty: Computer Security.
Semester: 09.
UE : UET9.
Subject title: Academic Communication and Research.
Credits: 1.
Coefficient: 1.

Teaching aim : The aim of this subject is to introduce students to the writing of scientific reports (articles, reports, theses, etc.) and the oral presentation of national and international scientific communications.

Recommended prerequisites: Mastery of the used language.

Contents :

- Principles of scientific communication.
- Publication modes: Article, Patent, Thesis, Book, Poster, Oral...
- Sources of full-text bibliographic information (the SNDL system, open access, archives, etc.).
- Structure of the different scientific publications (Articles, theses, oral presentation, etc.).
- Ethics in scientific research in Computer Science (Plagiarism, self-plagiarism, generative AI like chatGPT, etc.)
- Document preparation systems (LaTex) and bibliographic reference styles (APA, IEEEtran...etc).
- Bibliography management tools (Zotero, Mendely, EndNote, Bibtex...etc).

Assessment method: Exam.