

## DESCRIPTIONS ET BIBLIOGRAPHIES DES EXAMENS EN GENIE LOGICIEL

Groupe A		Groupe B	
19-LO-A1	Algorithmes et structures de données	19-LO-B1	Systèmes fiables
19-LO-A2	Architecture informatique et systèmes d'exploitation	19-LO-B2	Modélisation et vérification des logiciels (méthodes formelles)
19-LO-A3	Conception logicielle	19-LO-B3	Rétroingénierie, maintenance et évolution
19-LO-A4	Systèmes temps réel	19-LO-B4	Sécurité
19-LO-A5	Exigences et spécifications		
19-LO-A6	Assurance-qualité logicielle		
19-LO-A7	Processus de développement logiciel		

*La liste de manuels a été établie à seule fin d'aider les candidats à se préparer convenablement aux examens de l'Ordre. Cependant, étant donné qu'il peut arriver que ces manuels, ou les chapitres recommandés de ces manuels, ne couvrent pas toute la matière sujette à l'examen, nous rappelons aux candidats que **ce sont les descriptions des examens qui définissent la matière sur laquelle ils peuvent être interrogés et non les bibliographies.***

Règle générale, les manuels principaux couvrent la majeure partie ou la totalité de la matière des examens, alors que les manuels complémentaires peuvent servir de source additionnelle de référence. Lorsque deux ou plusieurs manuels principaux sont reliés par un "et", cela signifie que chacun de ces manuels ne couvre qu'une partie de la matière de l'examen.

Dans plusieurs cas, les chapitres (ch.) des manuels principaux couvrant la matière ou partie de la matière de l'examen sont indiqués, mais le candidat doit quand même vérifier si toute la matière de l'examen est bien couverte dans les chapitres en question. Lorsque tous les chapitres d'un manuel sont au programme, on indique "le manuel au complet". Enfin, l'absence de telles indications signifie que le candidat doit lui-même identifier les chapitres couvrant la matière définie dans les descriptions d'examens.

On notera que le (A) placé en marge des bibliographies indique le ou les manuels principaux recommandés en langue anglaise, alors que le (F) indique le ou les manuels principaux recommandés en langue française. L'absence de ces lettres signifie que le choix complet n'est pas offert dans les deux langues (à moins que cette possibilité soit indiquée autrement).

Les descriptions d'examens de 2019 demeurent en vigueur jusqu'à ce que le programme d'examens soit révisé. Les manuels recommandés peuvent cependant changer, soit parce qu'ils ont été mis à jour ou qu'ils ne sont plus disponibles. En général, il faut présumer que la plus récente édition d'un manuel recommandé peut remplacer l'ancienne édition. Dans le doute, il est conseillé de s'adresser au Service de l'accès à la profession : [examens\\_admission@oiq.qc.ca](mailto:examens_admission@oiq.qc.ca).

## EXAMINATION DESCRIPTIONS AND BIBLIOGRAPHIES FOR SOFTWARE ENGINEERING

Group A		Group B	
19-LO-A1	Algorithms and Data Structures	19-LO-B1	Dependable systems
19-LO-A2	Computer Architecture and Operating Systems	19-LO-B2	Software Modeling and Verification (Formal Methods)
19-LO-A3	Software Design	19-LO-B3	Reverse Engineering, Maintenance and Evolution
19-LO-A4	Real-Time Systems	19-LO-B4	Security
19-LO-A5	Requirements and Specifications		
19-LO-A6	Software Quality Assurance		
19-LO-A7	Software Development Process		

*The list of textbooks has been compiled solely to assist candidates in preparing for OIQ examinations. However, since these textbooks or recommended chapters may not completely cover the material on which the examination is based, we remind candidates that it is the examination descriptions that define what may be tested, not the bibliography.*

As a rule, the prime texts cover most or all the examination subject matter, while the supplementary texts may be used as additional sources. When two or more prime texts are linked by the word "et", it is an indication that each of these textbooks covers only a portion of the examination subject matter.

In a number of cases, chapters (ch.) of prime texts covering all or a portion of the examination subject matter are noted. However, candidates should check whether all the examination subject matter is in fact covered by the specific chapters. When all chapters of a textbook are relevant, the words "le manuel au complet" appear. Finally, the absence of such information means that the candidates must themselves locate the chapters covering the subject matter contained in the examination description.

The letter (A) appearing in the margin of the bibliography indicates the English prime texts recommended, while (F) indicates the French prime texts recommended. When neither letter appears, a full choice is not available in both languages (unless this possibility is indicated in some other way).

The descriptions of the 2019 examinations are valid until the examination program is revised. There may be changes in recommended textbooks, however, because there have been updates or the books are no longer available. It can be generally assumed that the latest edition of a recommended textbook replaces the older edition. In case of doubt, contact the Admission Department : [examens\\_admission@oiq.qc.ca](mailto:examens_admission@oiq.qc.ca).

**19-LO-A1**  
**ALGORITHMES ET STRUCTURES DE DONNÉES /**  
**ALGORITHMS AND DATA STRUCTURES**

**DESCRIPTION**

Structures de données fondamentales et algorithmes connexes. Piles et files d'attente, arbres, tables, listes, tableaux, chaînes, ensembles; fichiers et méthodes d'accès. Arbres balancés, structures multi-clés. Recherche. Tri. Techniques de conception d'algorithmes, telles que : diviser pour régner, méthode de l'algorithme glouton, équilibrage, programmation dynamique. Algorithmes se rapportant aux opérations ensemblistes. Graphes, algorithmes de graphes : recherche en profondeur d'abord et en largeur d'abord, arbre recouvrant de longueur minimum, algorithme du plus court chemin. Mesures empiriques et théoriques du rendement des algorithmes. Analyse de la complexité. Problèmes durs, problèmes NP-complets et problèmes insolubles.

Fundamental data structures and their associated algorithms. Stacks and queues, trees, tables, lists, arrays, strings, sets; files and access methods. B-trees, multi-key organizations. Searching. Sorting. Algorithm design techniques, such as divide and conquer, the greedy method, balancing, dynamic programming. Algorithms related to set operations, Graphs, graph algorithms: depth-first and breadth-first search, minimum spanning tree, shortest path. Empirical and theoretical measures of the efficiency of algorithms. Complexity analysis. Hard problems, NP-completeness, and intractable problems.

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2013-12-09)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

- (F) CORMEN, T., C. LEISERSON et R. RIVEST, Introduction à l'algorithmique. 2e éd., Paris, Dunod, 2002.
- (A) CORMEN, T., C. LEISERSON et R. RIVEST, Introduction to Algorithms. Third edition, Cambridge (MA), MIT Press, McGraw Hill, 2009.

**OR**

BRASSARD, G. and BRATLEY, P. Fundamental of Algorithmic. Englewood Cliffs (N.J.), Prentice-Hall, 1996.

**19-LO-A2**  
**ARCHITECTURE INFORMATIQUE ET SYSTÈMES D'EXPLOITATION /**  
**COMPUTER ARCHITECTURE AND OPERATING SYSTEMS**

**DESCRIPTION**

Éléments de base de l'architecture d'ordinateur, comprenant : algèbre booléenne, portes, logique combinatoire et séquentielle, représentation des données de niveau machine; organisation machine, programmation en langage assembleur/machine; organisation de la mémoire, caches, tas, piles; E/S série et parallèle, interruptions, protocoles de bus et accès direct en mémoire (DMA). Éléments de base des systèmes d'exploitation, comprenant : accès simultané, ordonnancement des processus, gestion de la mémoire; protection, accès et authentification; enchaînement et chargement; machines virtuelles.

Computer Architecture basics, including Boolean algebra, gates, combinational and sequential logic, machine-level representation of data; machine organization, assembly/machine language programming; memory organization, caches, heaps, stacks; serial and parallel I/O, interrupts, bus protocols, and direct memory access (DMA). Operating System basics, including concurrency, process scheduling, memory management; protection, access, and authentication; linking and loading; virtual machines.

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2020-12-18)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

- David A. Patterson, John L. Hennessy, Morgan Kaufman, Computer Organization & Design, 1998.
- David A. Patterson, John L. Hennessy, Computer Organization and Design MIPS Edition: The Hardware/Software Interface, 2013 edition.
- Saltzer and Kaashoek, Principles of Computer System Design: An Introduction.

**19-LO-A3  
CONCEPTION LOGICIELLE  
SOFTWARE DESIGN**

**DESCRIPTION**

Rôle de l'activité de conception logicielle. Attributs de qualité de la conception logicielle : exactitude, fiabilité, maintenabilité, portabilité, robustesse. Principes de conception logicielle : séparation d'aspects, abstraction, masquage d'information. Typage statique et dynamique. Types mutables et immuables. Modularité et décomposition. Conception orientée-fonction. Conception orientée-objet. Sous-typage. Éléments. Conception d'interfaces. Conception de niveau module. Notations : UML et autres notations. Concepts de base des schémas de conception. Introduction aux tests : tests unitaires, tests boîte noire c. tests boîte grise, couverture des tests.

Role of software design activity. Software design quality attributes: correctness, reliability, maintainability, portability, robustness. Software design principles: separation of concerns, abstraction, information hiding. Static and dynamic typing. Mutable and immutable types. Modularity and decomposition. Function-oriented design. Object-oriented design. Subtyping. Components. Interface design. Module level design. Notations: UML and other notations. Basic concepts of design patterns. Introduction to testing: unit tests, blackbox vs. grey box testing, test coverage.

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2007-03-14)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

**MANUELS PRINCIPAUX / PRIME TEXTS**

- (F)** UML 2 et les design patterns - Analyse et conception orientées objet et développement itératif, Larman, Craig, Pearson Education, 2005. ISBN 2-7440-7090-4.
- (A)** Applying UML and Patterns -- An introduction to Object-Oriented Analysis and Design and Iterative Development, 3rd Edition, Larman, Craig, Prentice Hall PTR, 2005 (703 pages).

**MANUELS COMPLÉMENTAIRES / SUPPLEMENTARY TEXTS**

Design Patterns, Erich Gamma , Richard Helm , Ralph Johnson , John Vlissides, Vuibert, 1999 (480 pages).

Design Patterns - Elements of Reusable Object-Oriented Software, Gamma, E., Helm, R., Johnson, R. et Vlissides, J., Addison-Wesley, 1995 (395 pages).

**19-LO-A4  
SYSTÈMES TEMPS RÉEL /  
REAL-TIME SYSTEMS**

**DESCRIPTION**

Définition et caractéristiques des systèmes temps réel. Systèmes temps réel « durs » et « doux ». Réactions dynamiques de processus physiques simples. Conception de systèmes temps réel (exigences, méthodes de conception, implémentation, tests, interaction personne-machine). Fiabilité et tolérance aux failles. Exceptions et traitement des exceptions. Accès simultané, synchronisation, communication et contrôle des ressources. Ordonnancement (exécutif cyclique, taux monotone et priorité des échéances, protocoles de plafond des priorités). Systèmes d'exploitation temps réel. Systèmes intégrés simples.

Definition and characteristics of real-time systems. Hard and soft real-time systems. Dynamic responses of simple physical processes. Designing real-time systems (requirements, design methods, implementation, testing, human-computer interaction). Reliability and fault tolerance. Exceptions and exception handling. Concurrency, synchronization, communication and resource control. Scheduling (cyclic executive, rate monotonic and deadline priority, priority ceiling protocols). Real-time operating systems. Simple embedded systems.

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2020-12-18)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

- Burns, Andy Wellings, Real-Time Systems and Programming Languages: Ada, Real-Time Java and C/Real-Time POSIX Addison –Wesely, 2009 ISBN-13: 978-0321417459
- Giorgio Buttazzo, Hard Real-Time Computing Systems, Springer ISBN 978-1-4614-0676-1

**19-LO-A5**  
**EXIGENCES ET SPÉCIFICATIONS /**  
**REQUIREMENTS AND SPECIFICATIONS**

<b>DESCRIPTION</b>	
<p>Sources et techniques d'obtention d'information. Paradigmes de modélisation, comprenant : modélisation de l'information, modélisation comportementale, modélisation du domaine, modélisation fonctionnelle, modélisation des contraintes. Exigences de qualité (par ex. : performance, convivialité, fiabilité, maintenabilité); expression des exigences de qualité de sorte qu'elles soient testables. Priorisation, analyse des options de remplacement, négociation, analyse des risques et analyse des impacts. Gestion des exigences, gestion de la cohérence, analyse des interactions, traçabilité. Documentation des exigences (par ex. : cas d'utilisation) et langages de spécification. Validation, examens et inspections, prototypage, validation des exigences non fonctionnelles. Conception d'essais de réception</p>	<p>Elicitation sources and techniques. Modelling paradigms, including information modelling, behavioural modelling, domain modelling, functional modelling, constraint modelling. Quality requirements (e.g., performance, usability, reliability, maintainability); expressing quality requirements so that they are testable. Prioritization, trade-off analysis, negotiation, risk analysis, and impact analysis. Requirements management, consistency management, interaction analysis, traceability. Requirements documentation (e.g., use cases) and specification languages. Validation, reviews and inspections, prototyping, validating</p> <p>non-functional requirements. Acceptance test design.</p>

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2011-01-06)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

Requirements Engineering : Fundamentals, Principles and Techniques, 1<sup>st</sup> Edition, Klaus Pohl, Springer, 2010. ISBN-10: 3642125778. ISBN-13: 978-3642125775.

**19-LO-A6  
ASSURANCE-QUALITE LOGICIELLE /  
SOFTWARE QUALITY ASSURANCE**

**DESCRIPTION**

<p>Concepts de validation et de vérification. Cycle de vie des logiciels et application de la validation et de la vérification. Processus d'assurance-qualité logicielle. Définitions de la qualité des produits logiciels. Caractéristiques de qualité. Définitions et spécifications de la qualité ingénierie. Définition et classification des défauts logiciels. Définitions de l'aptitude à l'usage et de la qualité du point de vue du client. Coûts des logiciels, coûts et économie de la qualité. Examens, revues de projet et inspections : concepts généraux, essais unitaires (module/ensemble), essais des sous-systèmes / tests d'intégration, tests de régression, tests basés sur les états, tests fonctionnels traditionnels, tests/analyses logiques, facteurs à considérer lors des tests orientés-objets (polymorphisme et héritage); analyse et tests de la sûreté/des défaillances.</p>	<p>Validation and verification concepts, software lifecycle and application of validation and verification, software quality assurance processes. Definitions of software product quality, quality characteristics, engineering quality definitions, specifications. Definition and classifications of software defects, fitness for use and customer quality definitions. Software costs, quality costs and economics. Reviews, walkthroughs and inspections. Unit (Module/Package) level testing, subsystem/integration testing, regression testing, state based testing, traditional functional testing, logical testing/analysis, OO testing considerations (polymorphism and inheritance). Safety/failure analysis and testing.</p>
---	--

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2014-02-13)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

**MANUELS PRINCIPAUX / PRIME TEXTS**

(F) L'assurance qualité logicielle 1 : Concepts de base, April Alain & Laporte Claude Y, LAVOISIER, Paris, 2011, ISBN 978-2-7462-3147-4

**ET**

L'assurance qualité logicielle 2 : Processus de support, April Alain & Laporte Claude Y, LAVOISIER, Paris, 2011, ISBN 978-2-7462-3222-8

(A) Guide to the Software Engineering Body of Knowledge (SWEBOK), Abran, A., Moore, J.W., Bourque, P., Dupuis, R., Chapter 11, Software Quality, Los Alamitos, CA, 2004. (Gratuit et disponible sur le web à l'adresse suivante : <http://www.computer.org/portal/web/swebok>)

**AND**

Software Quality Assurance - From theory to implementation, Daniel Galin, ISBN-13 978201709452

19-LO-A7  
PROCESSUS DE DÉVELOPPEMENT LOGICIEL/  
SOFTWARE DEVELOPMENT PROCESS

**DESCRIPTION**

Cycles de vie des logiciels. Modèles de processus logiciels. Contrôle et gestion du cycle de vie de logiciels corrects, fiables, maintenables et rentables. Documentation des logiciels. Outils de gestion de projet. Gestion des risques. Communication et collaboration. Cause et effets de l'échec d'un projet. Estimation des coûts et ordonnancement. Facteurs influant sur la productivité et le succès. Mesures de productivité. Gestion de la configuration. Gestion des défauts.	Software life cycles. Software process models. Control and life-cycle management of correct, reliable, maintainable and cost-effective software. Software documentation. Project management tools. Risk management. Communication and collaboration. Cause and effects of project failure. Cost estimation and scheduling. Factors influencing productivity and success. Productivity metrics. Configuration management. Defect management.
---	---

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2014-09-17)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

**MANUELS PRINCIPAUX / PRIME TEXTS**

L'assurance qualité logicielle 1, Concepts de base, Alain April et Claude Laporte, Hermes-Lavoisier, Paris, 2011, ISBN : 978-2-7462-3147-4, 386 pages.

Implementing the IEEE Software Engineering Standards, Michael E.C. Schmidt, SAMS Publishing, 2000, ISBN-10: 0672318571, ISBN-13: 978-0672318573, 256 pages.

**Normes et standards suivants / Standards and norms :**

**1) Normes / Norms**

ISO/IEC 12207 : 2008 Systems and software engineering — Software life cycle processes  
<http://www.iso.org/iso/home.html>

ISO/IEC 15504 : 2004 Information technology — Process assessment  
<http://www.iso.org/iso/home.html>

**2) Standards / Standards**

Capability Maturity Model Integration (CMMI) pour l'évaluation et l'amélioration des processus de génie logiciel

Pour le standard / For standard:

<http://cmminstitute.com/resource/cmmi-for-development-version-1-3/>

Version française:

<http://cmmiinstitute.com/resource/french-language-translation-of-cmmi-for-development-v1-3/>

English version :

<http://cmmiinstitute.com/resource/cmmi-for-development-version-1-3/>

### 3) Les processus / The Process :

<http://www.upedu.org/>

ET/AND

Software engineering process with the UPEDU, Robillard P.N., Kruchten P., with D'Astous, P., Addison Wesley, Pearson Education 2003. 346 p. Software engineering Process with the UPEDU, Robillard, Kruchten, D'Astous, Addison Wesley

N.B. : Le site concerne les Processus en génie logiciel : Les onglets sont les différents éléments des processus. Les onglets principaux sont : Disciplines, Role Set et Artifact sets.

The site covers the Software Engineering Process : The tabs represent the different elements of the process. The main tabs are : Disciplines, Role Set et Artifact sets.

**19-LO-B1  
SYSTÈMES FIABLES /  
DEPENDABLE SYSTEMS**

**DESCRIPTION**

<p>Défaillances logicielles et matérielles. Défaillances, défaillances latentes et pannes. Caractérisation des fonctions de défaillance, distribution de probabilité des défaillances, fonction de l'intensité des défaillances. Définition et mesures de la fiabilité des logiciels. MTTF, MTBF, MTTR, disponibilité, maintenabilité. Fiabilité matérielle et fiabilité logicielle. Techniques de prédiction des défaillances restantes, comprenant : injection de fautes, analyse de l'arbre de classification, couverture du code. Techniques générales du cycle de vie pour la production de logiciels fiables, comprenant : prévention des défauts; détection et élimination précoces des défauts; conception visant la robustesse; utilisation de mesures de processus; stabilisation des exigences, de la conception, du code et des artefacts d'essais. Détection des défaillances actives et passives. Programmation N-versions, points de reprise avant et arrière, blocs récupérateurs et techniques d'arbitrage. Traitement et correction des fautes, exceptions, tolérance aux fautes. Survivance, fonctions essentielles et fonctionnements en mode dégradé. Protection de l'intégrité des données.</p>	<p>Software and hardware faults. Faults, latent faults and failures. Characterization of failure functions, probability distribution of failures, failure intensity function. Software reliability definition and measures. MTTF, MTBF, MTTR, availability, maintainability. Hardware reliability and software reliability. Techniques for prediction of remaining faults, including fault injection, classification tree analysis, code coverage. General lifecycle techniques for producing reliable software, including defect prevention, early defect detection and removal; design for robustness; use of process measurements; stabilization of requirements, design, code and test artifacts. Active and Passive fault detection. N-version programming, forward and backward checkpointing, recovery blocks, and arbitration techniques. Fault handling and correction, exceptions, fault tolerance. Survivability, critical functions and degraded modes of operation. Data integrity protection.</p>
---	---

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2020-12-18)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

- Nancy G. Leveson, Engineering a Safer World: Systems Thinking Applied to Safety, 2012, MIT Press, ISBN-13: 978-0262016629.

19-LO-B2  
**MODÉLISATION ET VÉRIFICATION DES LOGICIELS  
(MÉTHODES FORMELLES) /  
SOFTWARE MODELING AND VERIFICATION (FORMAL METHODS)**

**DESCRIPTION**

Modélisation mathématique des logiciels, comprenant des sujets tels que la logique de programmation, l'algèbre de processus et les spécifications basées sur un modèle, les langages OCL et les spécifications algébriques. Raisonnement mathématique utilisant de tels modèles, dont les preuves d'exactitude du programme. Outils de contrôle statique de l'exactitude du logiciel par rapport à ses spécifications.

Mathematical modelling of software, including topics such as programming logics, process algebras, model based specification, object constraint languages, and algebraic specification. Mathematical reasoning using such models, including proofs of program correctness. Tools for static checking of the correctness of software relative to its specification.

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2020-12-18)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

- V. S. Alagar and K. Periyasamy, Springer-Verlag, Specification of Software Systems, 1998, ISBN 0-387-98430-5.

**19-LO-B3**  
**RÉTRO-INGÉNERIE, MAINTENANCE ET ÉVOLUTION /**  
**REVERSE ENGINEERING, MAINTENANCE AND EVOLUTION**

**DESCRIPTION**

Maintenance de logiciels : corrective, perfective et adaptative. Techniques de rétroingénierie de l'architecture et de la conception logicielles à des fins de compréhension des programmes. Réingénierie des systèmes et processus (technique et opérationnelle). Réusinage. Migration (technique et opérationnelle). Analyse d'impact. Gestion des versions et de la configuration. Modèles d'évolution des logiciels (théories, lois). Relations entre entités en évolution (par ex. : hypothèses, exigences, architecture, conception, code, jeux d'essais). Systèmes existants. Dette technique.

Software maintenance: corrective, perfective, and adaptive. Techniques for reverse engineering software architecture and design, for the purpose of program comprehension. System and process re-engineering (technical and business). Refactoring. Migration (technical and business). Impact analysis. Release and configuration management. Models of software evolution (theories, laws). Relationship among evolving entities (e.g., assumptions, requirements, architecture, design, code, test suites). Legacy systems. Technical debt.

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2010-02-25)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

**MANUELS PRINCIPAUX / PRIME TEXTS**

Guide to the Software Engineering Body of Knowledge (SWEBOK), Abran, A., Moore, J.W., Bourque, P., Dupuis, R., Chapter 6 - Software Maintenance, Los Alamitos, CA, 2004. Visitez le site internet à l'adresse suivante : <http://www.computer.org/portal/web/swebok>.

**ET / AND**

Software Maintenance Concepts and Practice, 2<sup>nd</sup> Edition, Grubb, P., Takang A., World Scientific Publishing, 2003.

**MANUELS COMPLÉMENTAIRES / SUPPLEMENTARY TEXTS**

Software Engineering - Software Life-Cycle Processes – Maintenance, ISO/IEC 14764, ISO and IEC, 2006.

Software Maintenance Management : Evaluation and Continuous Improvement, April, A., Abran, A., John Wiley & Sons, 2008.

**19-LO-B4  
SÉCURITÉ /  
SECURITY**

**DESCRIPTION**

Caractéristiques des problèmes de sécurité. Chiffrement et déchiffrement de base; chiffrement sécurisé, protocoles et pratiques de chiffrement. Systèmes cryptographiques à clés publiques et signatures numériques. Conception logicielle visant la sécurité. Sécurité des systèmes d'exploitation, des bases de données, des ordinateurs personnels, des réseaux informatiques et des communications électroniques. Analyse des risques et planification de la sécurité. Dimension éthique de la sécurité informatique.

Characteristics of security problems. Basic encryption and decryption; secure encryption, encryption protocols and practices. Public key cryptosystems and digital signatures. Software design for security. Security in operating systems, databases, personal computers, computer networks and electronic communications. Risk analysis and security planning. Ethical issues in computer security.

**BIBLIOGRAPHIE / BIBLIOGRAPHY (Dernière mise à jour / Last update : 2015-02-18)**

***L'ÉDITION LA PLUS RÉCENTE EST RECOMMANDÉE/MOST RECENT EDITION IS RECOMMENDED***

**MANUEL PRINCIPAL / PRIME TEXT**

Security in Computing, Charles P. Pfleeger, Shari L. Pfleeger, Jonathan Margulies, 5th Edition (2015)  
ISBN-10: 0-13-408504-3, ISBN-13: 978-0134085043

**MANUELS COMPLÉMENTAIRES / SUPPLEMENTARY TEXTS**

Computer Security: Principles and Practice, William Stallings, Lawrie Brown, 3rd Edition (2014)

Sécurité informatique et réseaux: Cours avec plus de 100 exercices corrigés, Solange Ghernaoui  
4ème Édition (2013)

Cryptography Engineering: Design Principles and Practical Applications Niels Ferguson, Bruce Schneier, Tadayoshi Kohno, (2010)

Building Secure Software: How to Avoid Security Problems the Right Way, John Viega, Gary McGraw (2002)